

中國文化大學
資訊工程學系
資訊系統專題

人工智慧與物聯網 APP

學 生：吳梓境

吳承諺

蔡旻軒

紀俊丞

指導教授：周立平

中華民國 107 年 6 月

人工智慧與物聯網 APP

專題學生：吳梓境、吳承諺、蔡旻軒、紀俊丞

指導教授：周立平 博士

中國文化大學 資訊工程學系

摘要

隨者資訊日新月異，大家都是人手一機的情況下，總是希望能結合所有東西在一支手機，包刮了冷氣，電視，電腦等等，若是能將所有家電的控制都以手機來完成，那就方便非常多，不會再出現遙控器找不到的窘境，早上喝咖啡習慣的人，也能提早設定煮開水的時間，不須另外等待，回家路上也能先開好冷氣..等等，我們的專題主題就是以此方向開始研究，我們希望做出以手機 APP，就能夠連接所有家中所會用到的任何店子商品，以達到控制的目標。即時影像也是我們想呈現的一項功能，本專題所開發的人工智能與物聯網 APP 實際就是應用於一班家庭中，完整的建立出一套家庭物聯網系統，利用本專題的成果，在未來能夠給更多家庭帶來更方便的生活方式。

關鍵詞：人工智能、物聯網、智慧家庭

指導教授_____（簽名）

Artificial Intelligence(AI) Combined with Internet of Things (IoT) Application

**Student: Wu-Tzy Jing, Chen-Yen Wu, Min-Shiuan Tsai and
Gun-Chen Gi**

Advisor: Prof. Li-Ping Choui

**Department of Computer Science and Information Engineering
Chinese Culture University**

ABSTRACT

With the rapid advancement of information, we all hope that we can combine everything in a mobile phone, as if air condition, television, computer, etc. If we can control all our house appliances just through our mobile phone, that would be really convenient. There will no longer be a dilemma that cannot find your remote controller. People who drink coffee in the morning can also set the time for boiled water earlier without having to wait any longer. Having the air condition turned on on the way home, and no need to wait for the room to cool down, And so on, our thematic is to start research in this direction. We hope to make a mobile phone app that will be able to connect all the house appliances that will be used in all homes in order to achieve the goal of control. The real-time image is also a feature that we want to present. The artificial intelligence and the Internet of Things APP developed on this topic, we hope our app is applied to ordinary families and complete a set of home IoT system. The results of this topic will be used in the future. We're sure it Can bring much more convenient to more families.

Keywords: Artificial Intelligence 、 Internet of Things 、 Smart Home

目 錄

中文摘要.....	I
ABSTRACT.....	II
表 目 錄.....	V
圖 目 錄.....	VI
第 1 章 研究動機與目的	1
1.1 研究動機	1
1.2 研究目的	1
第 2 章 文獻探討.....	2
2.1 物聯網	2
2.2 語音識別	2
2.3 APP 開發平台 ANDROID STUDIO.....	3
2.4 樹莓派	4
2.5 各種感測器	5
第 3 章 研究內容與方法	7
3.1 軟體發展設計	7
3.1.1 簡介.....	7
3.1.2 系統概述	8
3.2 軟體系統架構	11
3.2.1 手機 APP 系統.....	11
3.2.2 人工智慧系統	18
3.2.3 樹莓派系統	30
3.3 硬體系統架構	31
3.3.1 系統硬體設備	31
3.3.2 物聯網系統架構	31
第 4 章 初步實驗結果與人力配置	33
4.1 研究進度	33

4.1.1 手機 APP 部分	33
4.1.2 人工智慧部分	41
4.1.3 物聯網部分	47
4.2 研究人員分配	48
第 5 章 預期成果.....	49
參考文獻.....	50

表 目 錄

表 4.1 參與本項研究之人員及工作內容	48
----------------------------	----

圖 目 錄

圖 3.1 系統架構示意圖	10
圖 3.2 智慧家庭系統架構示意圖	11
圖 3.3 Photoshop 介紹.....	12
圖 3.4 Flow 介紹	13
圖 3.5 Sketch 介紹.....	13
圖 3.6 APP 功能示意圖	16
圖 3.7 APP 頁面架構圖	17
圖 3.8 語音辨識環境建置圖	19
圖 3.9 Wit.AI 文字處理訓練流程圖	20
圖 3.10 AI 訓練圖.....	21
圖 3.11 AI 訓練圖.....	21
圖 3.12 AI 訓練圖.....	22
圖 3.13 AI 訓練圖.....	22
圖 3.14 AI 訓練圖.....	23
圖 3.15 AI 訓練圖.....	23
圖 3.16 AI 訓練圖.....	24
圖 3.17 AI 訓練圖.....	24
圖 3.18 AI 訓練圖.....	25
圖 3.19 AI 訓練圖.....	25
圖 3.20 AI 訓練圖.....	26
圖 3.21 AI 訓練圖.....	26
圖 3.22 AI 訓練圖.....	27
圖 3.23 AI 訓練圖.....	27
圖 3.24 AI 訓練圖.....	28
圖 3.25 AI 訓練圖.....	28
圖 3.26 AI 訓練圖.....	29

圖 3.27 AI 訓練圖.....	29
圖 3.28 AI 訓練圖.....	30
圖 3.29 Raspbian.....	30
圖 3.30 物聯網系統概念示意圖	32
圖 4.1 APP 登入頁面	34
圖 4.2 APP 主控制頁面	35
圖 4.3 APP 新增區域頁面	36
圖 4.4 APP 區域配置頁面	37
圖 4.5 APP 裝置設定頁面	38
圖 4.6 APP 選單頁面	39
圖 4.7 語音辨識頁面	40
圖 4.8 node 環境設置圖.....	41
圖 4.9 node 環境設置圖.....	42
圖 4.10 node 環境設置圖.....	42
圖 4.11 node 環境設置圖.....	43
圖 4.12 node 環境設置圖.....	43
圖 4.13 node 環境設置圖.....	44
圖 4.14 node 環境設置圖.....	44
圖 4.15 node 環境設置圖.....	45
圖 4.16 node 環境設置圖.....	45
圖 4.17 node 環境設置圖.....	46
圖 4.18 node 環境設置圖.....	46
圖 4.19 Raspberry pi 運行截圖	47

第 1 章 研究動機與目的

1.1 研究動機

在這科技越來越發達的時代，人們不可省略的問題就是改善工作上的效率與品質，為了改變人在勞動力上付出的更少，進而開發出一套物聯網產品。為了使農民或是各種重複性的大量工作者使他們方便管理而使用所學之技術來加入改善。

1.2 研究目的

使用我們目前所學以及所知的技術來達成我們的目標，第一步是使用樹莓派串起各式感測器與感測元件使我們可以輸入我們的程式碼來使用，第二步是撰寫一個 APP 頁面，第三步是加入人工智慧的技術，最終在將所有的技術呈現在 APP 的上面完成我們的需求。

(1) 使用樹梅派

加入各式感測器與感測元件，例如：溫度感測器、壓力感測等。使我們可以使用感測器的特性來做到我們需要達成的目的。

(2) APP 開發

使用 Android studio 當開發環境，之後使用它與樹梅派和人工智慧與語音辨識作結合使它達到我們需求。

(3) 使用人工智慧

使用語音辨識系統，在某些特定的環境或情況下可能有時候無法使用到肢體來使用裝置，那這時我們就需要語音辨識系統來讓人們更方便可以使用裝置。

第 2 章 文獻探討

2.1 物聯網

物聯網（英語：Internet of Things，縮寫 IoT）是網際網路、傳統電信網等資訊承載體，讓所有能行使獨立功能的普通物體實現互聯互通的網路。物聯網一般為無線網，而由於每個人周圍的裝置可以達到一千至五千個，所以物聯網可能要包含 500 兆至一千兆個物體。在物聯網上，每個人都可以應用電子標籤將真實的物體上網聯結，在物聯網上都可以查出它們的具體位置。通過物聯網可以用中心電腦對機器、裝置、人員進行集中管理、控制，也可以對家庭裝置、汽車進行遙控，以及搜尋位置、防止物品被盜等，類似自動化操控系統，同時透過收集這些小事的資料，最後可以聚整合大資料，包含重新設計道路以減少車禍、都市更新、災害預測與犯罪防治、流行病控制等等社會的重大改變，實現物和物相聯。

物聯網將現實世界數位化，應用範圍十分廣泛。物聯網拉近分散的資訊，統整物與物的數位資訊，物聯網的應用領域主要包括以下方面：運輸和物流領域、健康醫療領域範圍、智慧型環境（家庭、辦公、工廠）領域、個人和社會領域等，具有十分廣闊的市場和應用前景。

[6]使我們了解到物聯網這項技術再為來世不可避免的，無論是物療或醫療等，都將會使用到物聯網，這也是我們這次專題決定加入物聯網技術的原因。

2.2 語音識別

語音識別（speech recognition；語音辨識／言語辨別）技術，也被稱為自動語音識別（英語：Automatic Speech Recognition, ASR）、電腦語音識別（英語：Computer Speech Recognition）或是語音轉文

本識別（英語：Speech To Text, STT），其目標是以電腦自動將人類的語音內容轉換為相應的文字。與說話人識別及說話人確認不同，後者嘗試識別或確認發出語音的說話人而非其中所包含的詞彙內容。

語音識別技術的應用包括語音撥號、語音導航、室內設備控制、語音文檔檢索、簡單的聽寫數據錄入等。語音識別技術與其他自然語言處理技術如機器翻譯及語音合成技術相結合，可以構建出更加複雜的應用，例如語音到語音的翻譯。

這方面我們了解到語音識別技術所涉及的領域包括：信號處理、模式識別、機率論和資訊理論、發聲機理和聽覺機理、人工智慧等等。

語音識別這部分我們決定透過 Wit.Ai 語音辨識網站來達成語音識別的部分，我們會先將錄音檔傳送給 Wit.Ai，在該網站作分析以後，再相分析結果回傳以供我們後續的使用。[7]

2.3 APP 開發平台 ANDROID STUDIO

Android 嵌入式作業系統(open structure)，目前用於手機最多，所以大家都只認為它是手機作業系統，以 Linux 為基礎所開發的 OS。適用於：手機、PDA、MID、Notebook、嵌入式設備(家電、穿戴式裝置)等等。可跨平台開發(iPhone 除外)：Windows、Linux、MAC。

Google 提供開發 Android App 的套件。Android 不同版本間因為系統有更修正，所以每個版本都有獨立的 SDK，用於程式開發與建立模擬器，Android 版本越大，SDK 也越大，意味著電腦容量與等級也要跟著提高，執行效率會較佳。預設只安裝最新的 SDK，需要時才更新，如果沒有安裝對應的 SDK，則無法使用該版本的 Android 功能

選擇 Android Studio 的主要原因為我們都修過該堂課，使用起來較熟悉，也考量到他能夠跨平台開發的優點，使我們決定使用該款軟體完成我的 APP 的手機端。[8]

2.4 樹莓派

樹莓派 (Raspberry pi) 是一款由英國的樹莓派基金會所開發 Linux 單機板電腦。樹莓派只有一張信用卡大小，可執行 ARM 架構的 Debian、Arch Linux 和 Fedora 等的發行版作業系統，支援 Python 作為主要程式語言，亦支援 BBC BASIC、C 語言和 Perl 等程式語言。

Raspberry Pi 3 搭載全新 64 位元四核心 ARM Cortex-A53 (1.2GHz)，具有 4 組 USB 2.0、10/100 乙太網路，更支援無線網路與藍芽。由於體積不大耗電量低的特點，非常適合做為感測器節點和物聯網的節點主機，透過它可以將資料儲存於本地端，亦能同步上傳到雲端伺服器中。

樹莓派方面我們最後決定使用 Raspberry Pi 3 model B+，原因是他是樹莓派最新一代板子，內建了無線網路模組以及藍芽模組。我們也將架設外網伺服器讓使用者可以不受空間限制使用該項裝置。[9]

2.5 各種感測器

本研究將使用多種感測器，以下說明選用所需之各種感測器。

(1) 影像感測器

可使用影像來進行外觀檢查、文字檢查、定位、不良品檢查等。

1) 智慧型攝影機 FQ-2

檢測能力、攝影機種類、通訊種類等皆獲得大幅進化的全新智慧型攝影機。具備高階影像感測器的功能，「無需控制設備的影像感測器」。令顧客優先考慮「優先選用 FQ2」的全新智慧型攝影機。[5]

(2) 一氧化碳感測器

用於家庭、環境的一氧化碳探測裝置。適宜於一氧化碳、煤氣等的探測。MQ-7 氣體傳感器所使用的氣敏材料是在清潔空氣中電導率較低的二氧化錫(SnO_2)。採用高低溫循環檢測方式低溫（1.5V 加熱）檢測一氧化碳，傳感器的電導率隨空氣中一氧化碳氣體濃度增加而增大，高溫（5.0V 加熱）清洗低溫時吸附的雜散氣體。使用簡單的電路即可將電導率的變化，轉換為與該氣體濃度相對應的輸出信號。MQ-7 氣體傳感器對一氧化碳的靈敏度高，這種傳感器可檢測多種含一氧化碳的氣體，是一款適合多種應用的低成本傳感器。[5]

(3) 煙霧感測器

煙霧感測器是一種探測煙霧的儀器，一般用來預警火災，或在一些禁菸的場所。煙霧探測器一般存放於塑膠製的圓形外殼之中。大部份煙霧探測器利用光電效應或電離的原理偵測煙霧，而需要較高靈敏度、用來偵測吸菸的煙霧探測器則同時包含這兩種偵測方法的儀器。獨立住宅中常用的煙霧探測器多為獨立運作，以乾電池作為供電方式，並在偵測到煙霧時自行發出聲響。[5]

(4) 智慧插座

隨著物聯網（Internet of things，IoT）越趨流行，智慧插座順應而生。許多家電已經智慧化，可以定時開關機，省電又符合使用者的需求，然仍有許多家電無法達到需求，智慧插座變順應而生。智慧插座只需連接在普通插座及電器插頭中間即可。透過插座中的晶片連結上網路或手持裝置，即可完成安裝。[5]

第 3 章 研究內容與方法

3.1 軟體發展設計

本研究透過國內外討論之手機端應用程式與語音辨識 AI 之文獻，架構出新的一套語音辨識與物聯網應用的 APP。

3.1.1 簡介

本研究以市面上諸多廠牌之智慧家庭系統與國內外學術討論網站為案例，開發出一套自製的語音辨識與控制之手機 APP。分為目的、計畫範圍、物聯網系統、人工智慧系統、APP 系統。

(1) 目的

本研究以改善人類生活、提升工作效率為目的，利用物聯網的方式達到隨時隨地進行操作、監控等用途，使人類生活智慧化。

(2) 計畫範圍

本研究分為三大部分，人工智慧、手機 APP 與樹梅派和物聯網系統，由人工智慧進行語音辨識，再由手機 APP 連線樹梅派對已安裝之物聯網設備進行操作與控制。

(3) 物聯網系統

物聯網是繼互聯網之後下一代網路發展方向，核心概念是將人們周遭的事物透過各種通信手段連入網路中以達到感知與控制的目的，智慧家庭是物聯網系統研究的一個子研究方向，透過將環境中各種電子設備連網，達成用戶可以遠程控制與感知環境進而提供使用者更便利更安全的生活環境。

(4) 人工智慧系統

人工智慧是近年來科技發展的主流，透過深度學習建構出對於每個獨立問題的特殊方程式，從而獲得自主應對與解析的能力。

語音辨識是人工智慧研究的一個子方向，透過大量訓練得以理解每個音節結合在一起形成字句，使語音轉為文字(STT)變為可能。

(5) 手機 APP 系統

手機做為現代人最常使用的電子設備，基本上已經成為了現在人有如貼身衣物一般的存在，因此本研究採用手機 APP 結合物聯網系統配合語音辨識系統，實現人們更便利、更有效率的生活。

3.1.2 系統概述

本文件旨在說明「人工智慧與物聯網 APP」之系統目標、系統範圍、系統架構、軟體架構。

(1) 系統目標

現今社會上科技設備使用的越來越普遍，手機的使用程度已經超越了許多貼身裝置，每個人都有個需求，希望能夠遠端掌握一切資訊甚至於操作，忙碌的上班族總會希望一下班回到家，等待著自己的就是舒適的環境，在回家前打開冷氣一進門就享受著涼爽，出遠門不便帶寵物的家庭，透過遠端操作監視器看看家中的寵物是否依然健康安全等，實現人類更舒適的生活就是本研究的目標，給人類更便利更友善的環境。

(2) 系統範圍

本文件旨在說明本系統範圍，包含系統名稱、系統功能說明、系統架構、系統建構項目需求概述及軟、硬體需求環境。

1) 系統名稱

Android 手機應用程式—E 能宅

2) 系統功能說明

A. 物聯網端說明

手機 APP 透過乙太網路連線登入樹莓派，再由樹莓派藉由 Wi-Fi 連接傳感器進行即時監控。

B. 手機 APP 端說明

[a] 會員登入

提供使用者登入樹莓派產生連線。

[b] 設備訊息

各種傳感器的即時資訊、開關設定調整。

[c] 設定

新增裝置、刪除裝置、定時模式等功能調整。

[d] 使用介紹

提供使用者簡易的教學，如何透過手機 APP 對已連網設備進行調整、檢視。

C. 人工智慧端說明

利用 wit.ai 提供的 API 套件進行語音辨識，語音指令透過客戶端 APP 傳至 wit.ai，解析後回傳，根據回傳解析之指令執行。

(3) 系統架構

本系統分成三個子系統開發，分別有手機用戶端 APP、物聯網系統、人工智慧系統，三個子系統相互連結形成本系統，如圖 3.1。

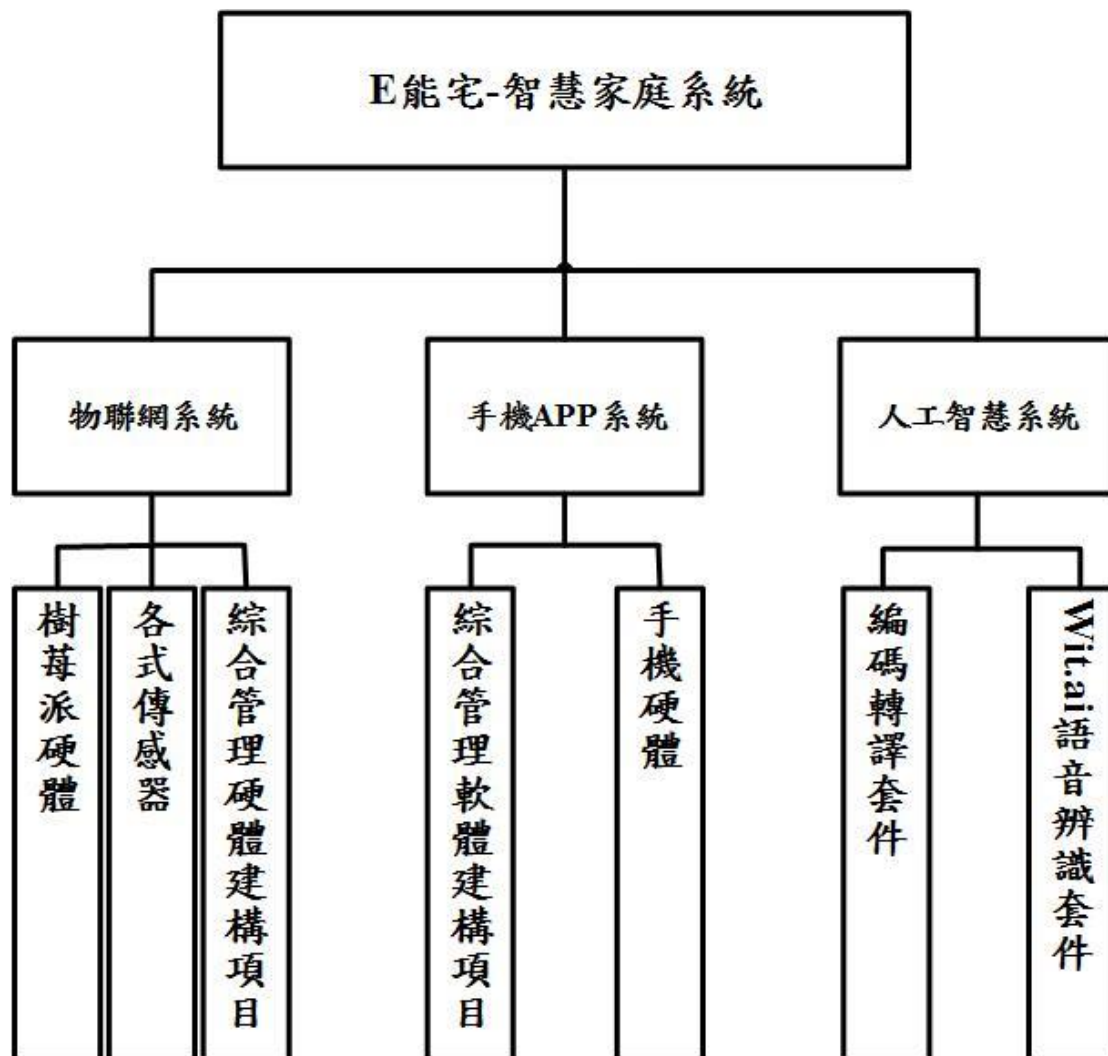


圖 3.1 系統架構示意圖

(4) 測試環境架構

測試環境設計架構上採取精簡高效的方式，考量各種家中、辦公室會出現的變數，設計簡便的測試環境架構。如圖 3.2。

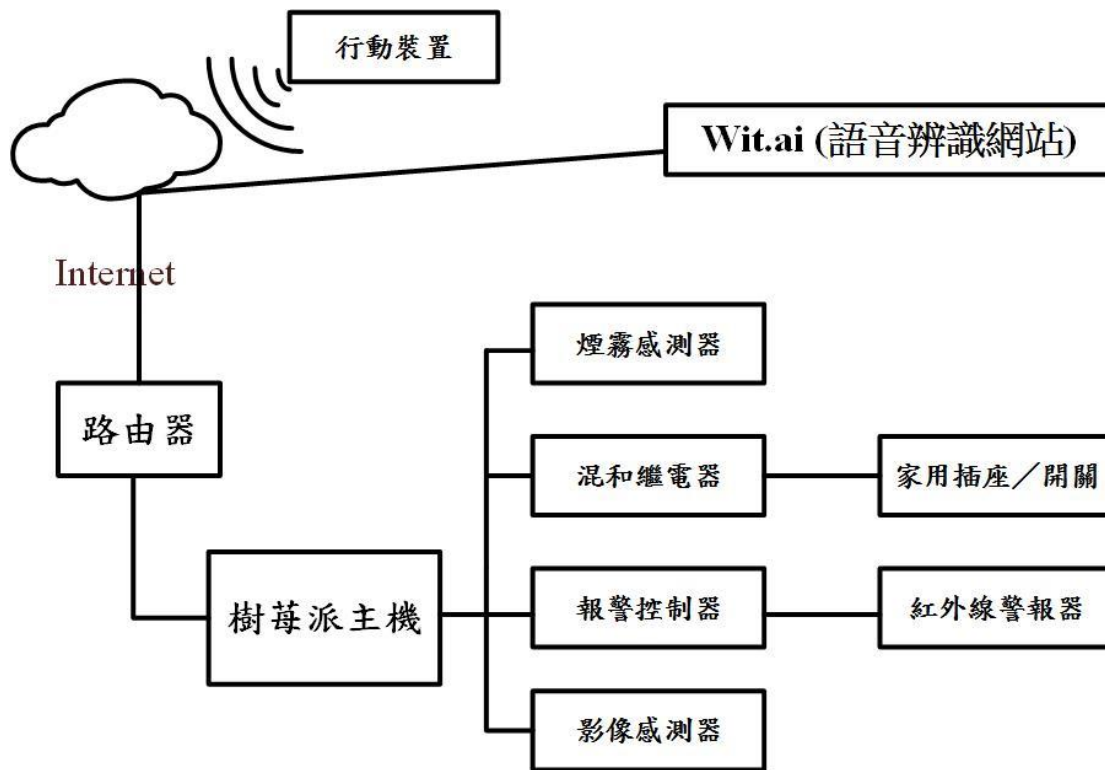


圖 3.2 智慧家庭系統架構示意圖

3.2 軟體系統架構

本節將說明手機 APP 系統、人工智慧系統與樹莓派系統所需之驅動、開發環境與相對應之功能。

3.2.1 手機 APP 系統

本節將說明手機 APP 系統之開發環境、APP 所含之所有功能與 APP 頁面架構。

(1) 開發環境

本節將闡述本研究之 APP 系統使用的開發環境與軟體。

1) Photoshop

主要是與 in vision 一同使用，在 Photoshop 設計好的圖示，能夠直接匯入 In vision 並且設定成獨立的物件，以供我們調整位置大小等等，如圖 3.3。

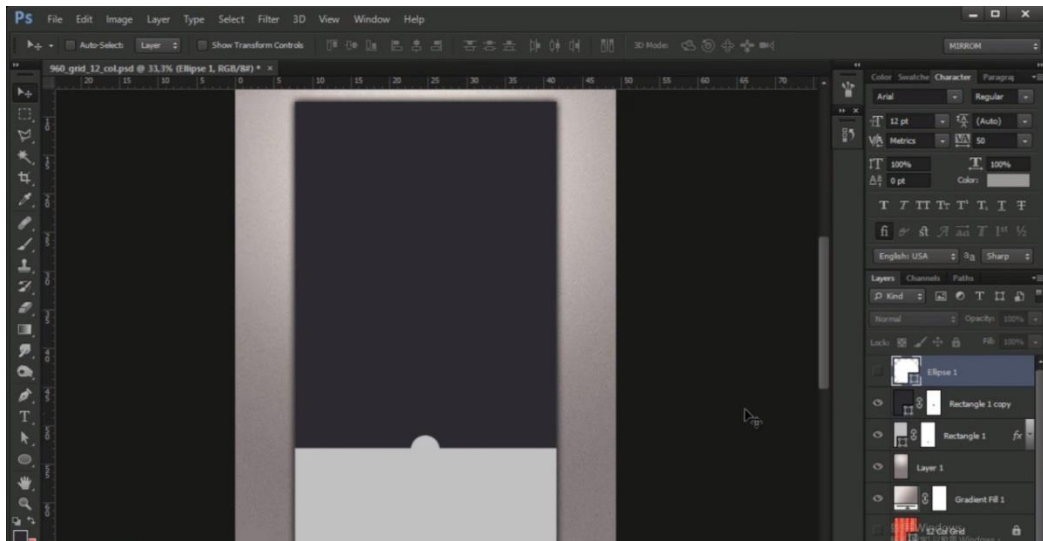


圖 3.3 Photoshop 介紹

2) Flow

是我們將在 APP 中呈現出動態化石所會用到的程式，我們希望能在所有開關按鈕上都使用這項功能，達到一個動態呈現，如圖 3.4。

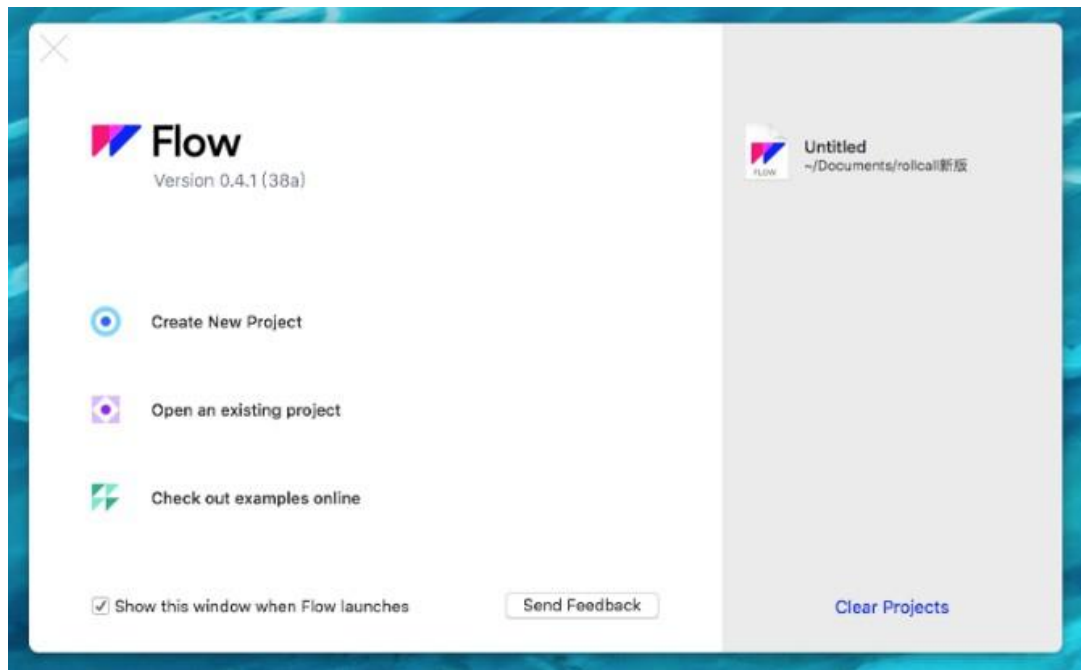


圖 3.4 Flow 介紹

3) Sketch

Sketch 則是手機 APP UI 設計上最推薦的軟體了，唯一缺點是只能在 IOS 系統上座使用，因此這一款我們目前還在考慮中，如圖 3.5。

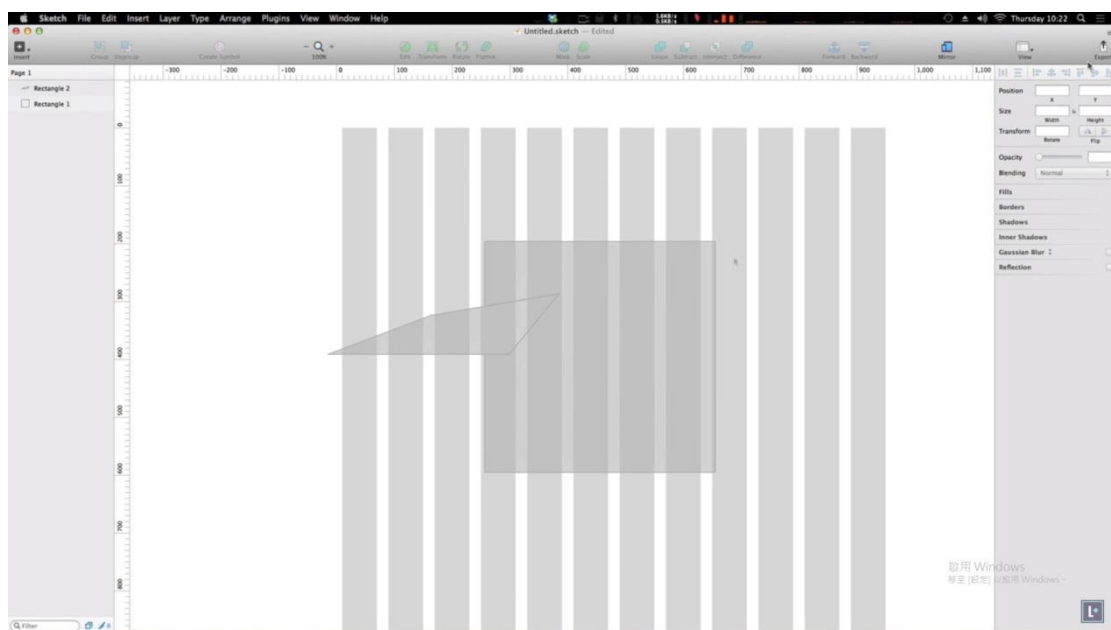


圖 3.5 Sketch 介紹

(2) APP 功能

以下將說明本 APP 內所有功能與相對應的頁面配置，如圖 3.6。

1) 登入系統

當使用者輸入帳號密碼時即可登入至樹莓派系統使用該系統。

2) 新增區域

區域系統可以新增區域，有條理的規劃使用空間，方便管理眾多數量的裝置。

3) 刪除區域

刪除不必要的區域。

4) 新增設備

可以新增我們有支援的裝置作連結，例如：影像控制器、煙霧感測器等。其他種類的控制器與感測器。

5) 刪除設備

當使用者不需要或是此設備有問題時可以使用此功能刪除選擇的設備。

6) 查看設備

使用者可以查詢設備，適合給新手使用與剛新增此設備的使用者未它做深入的介紹。

7) 裝置設定

在這個選項中使用者可以依喜好設定它所適合此應用之感測器的數值，例如每個用戶的喜好不同所喜歡的燈光亮度也不一樣，就可以在此調整喜愛的亮度素質。

8) 執行模式

可以調整喜愛與所需的模式，例如我今天只需要它運作三小時就可以使用定時模式等等。

9) 語音操控

開啟語音操控的模式，一個方便使用者操作的功能。

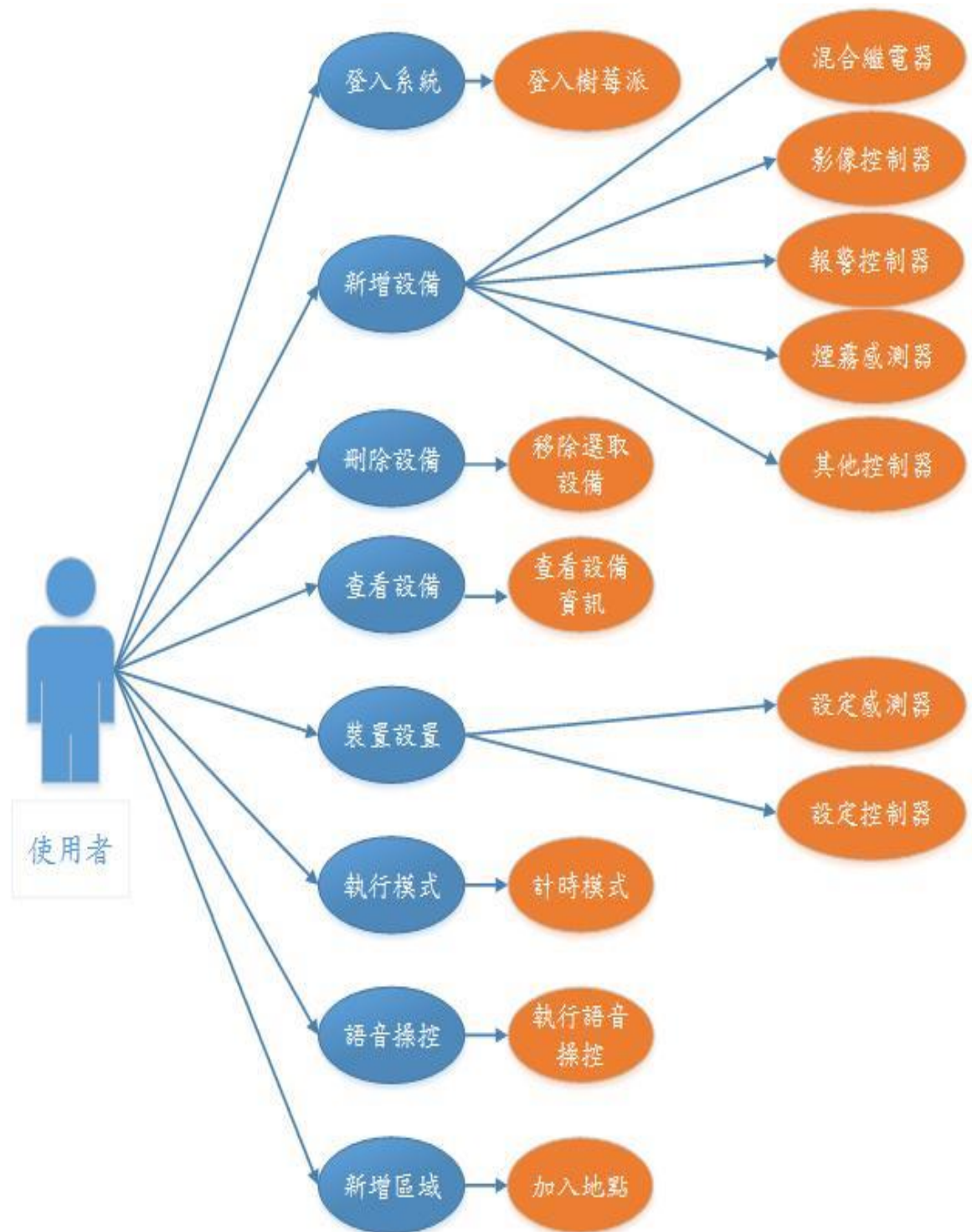


圖 3.6 APP 功能示意圖

(3) APP 頁面架構

以下將說明本 App 所有頁面與頁面之間互動關係。

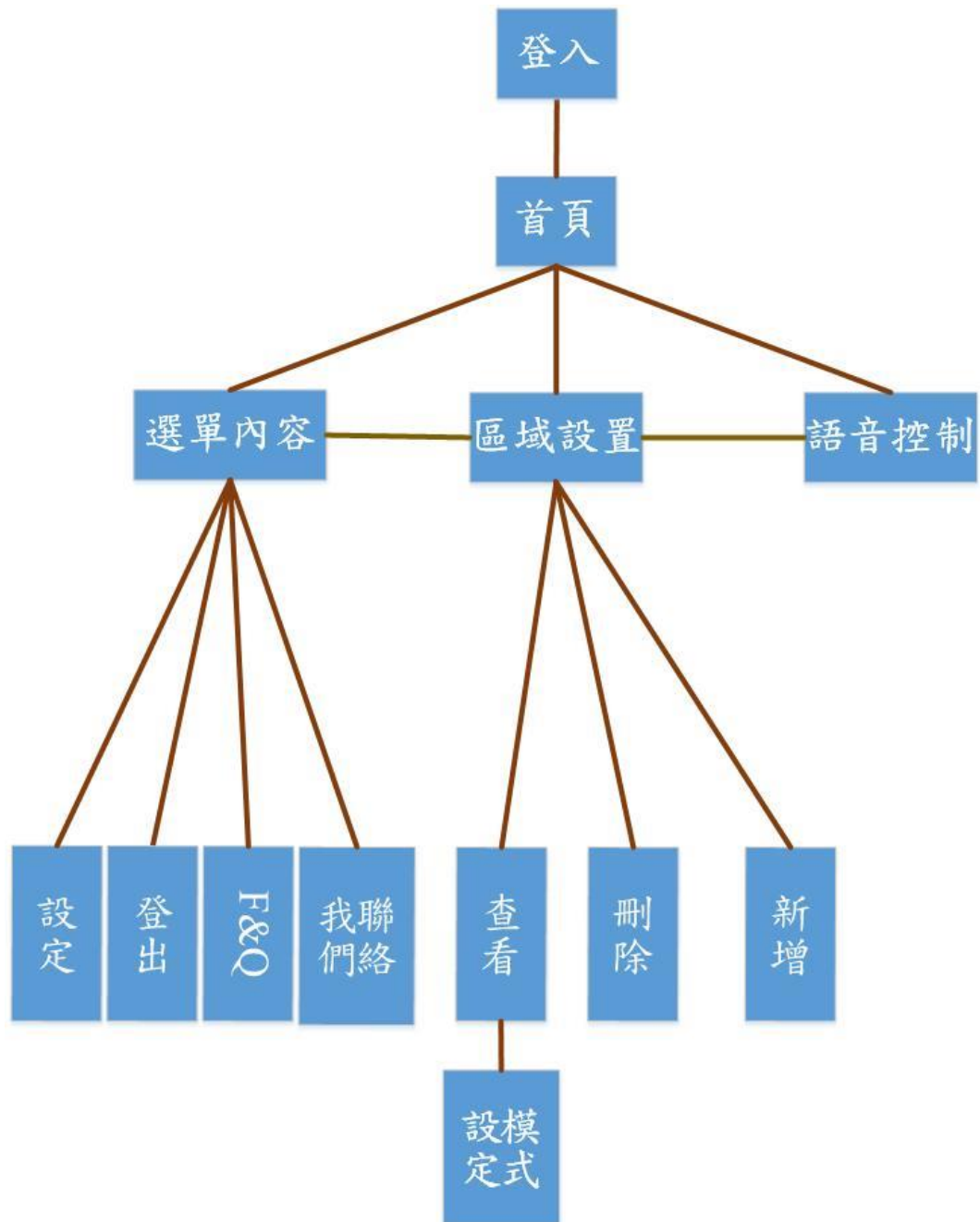


圖 3.7 APP 頁面架構圖

3.2.2 人工智慧系統

採用 Wit.ai 的開源語音套件作為語音辨識的軟體，以下將介紹語音辨識系統音訊轉換文字流程與文字處理訓練流程，並以圖片的方式詳細介紹文字處理部分訓練步驟。

(1) Wit.ai 語音辨識系統環境建置

為了進行語音辨識，需先透過 APP 收集音訊，將此音訊傳送至 Wit.ai，如圖 3.8。

接著才進行文字處理訓練並將訓練完成的成果以 API 的形式套用進 APP 系統，如圖 3.9。

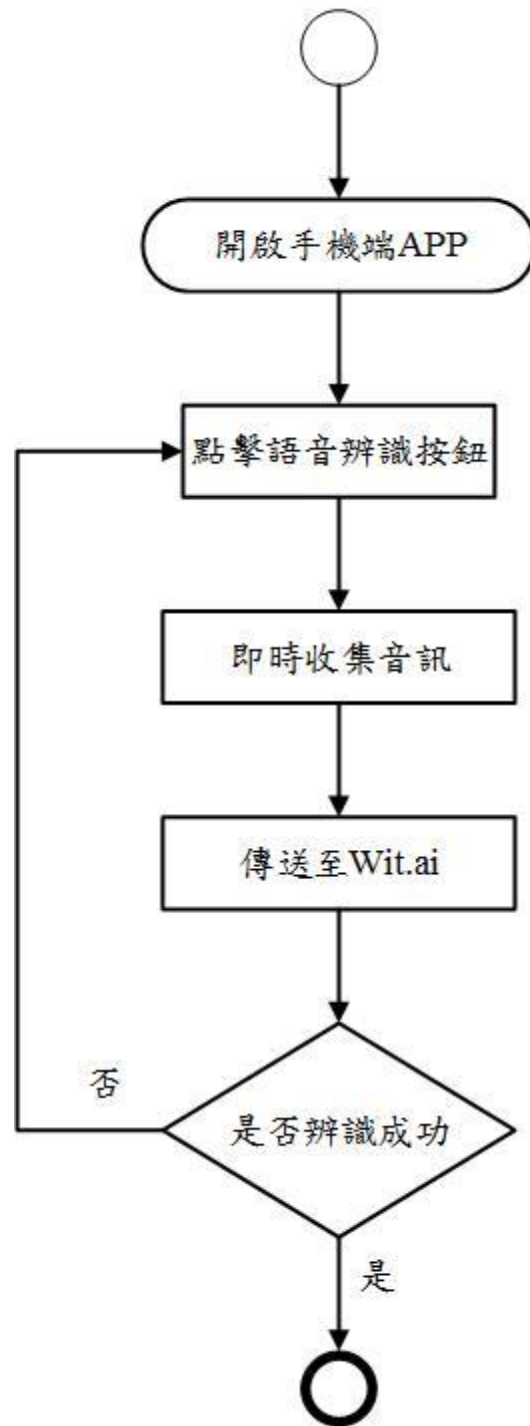


圖 3.8 語音辨識環境建置圖

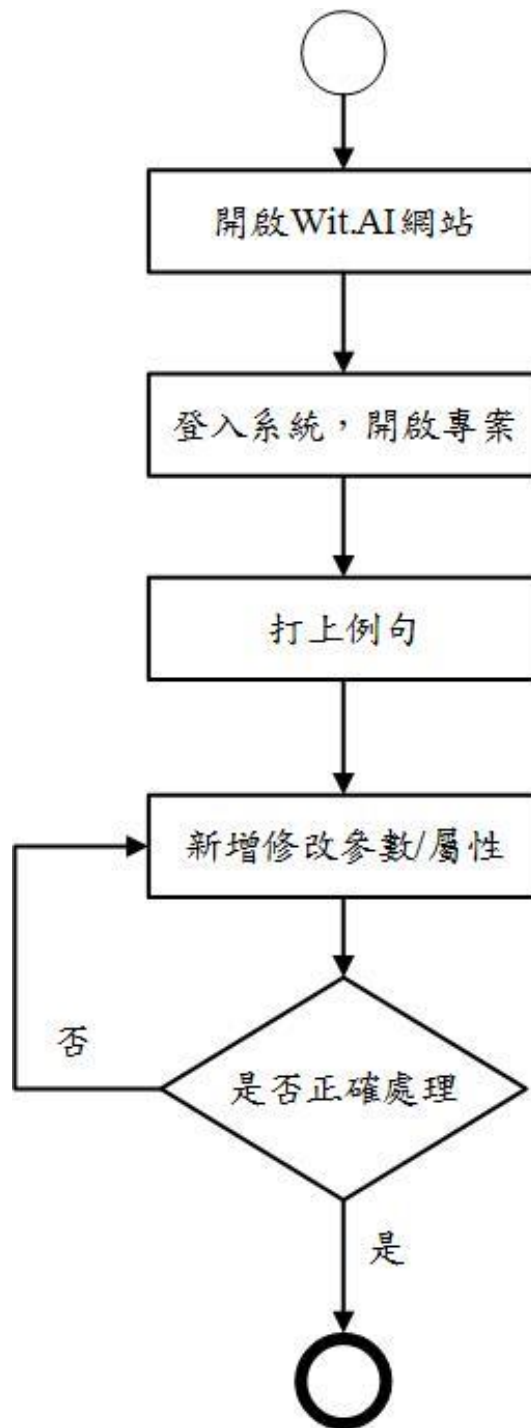


圖 3.9 Wit.AI 文字處理訓練流程圖

(2) Wit.ai 文字處理訓練流程

本節將說明如何訓練 Wit.ai 進行文字辨識，訓練完成後方可以 API 的形式加入本研究之 APP 系統進行語音辨識。

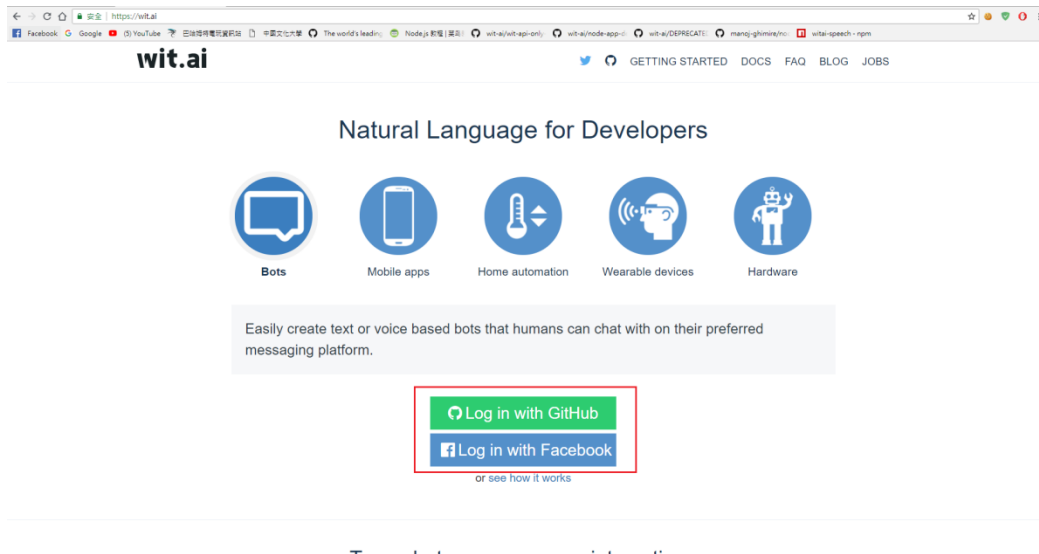


圖 3.10 AI 訓練圖

步驟 1.前往 Wit.Ai 網站(網址：http://wit.ai)登入介面，於紅色框框處可選擇 GitHub 帳號或 Facebook 帳號登入，如圖 3.10。

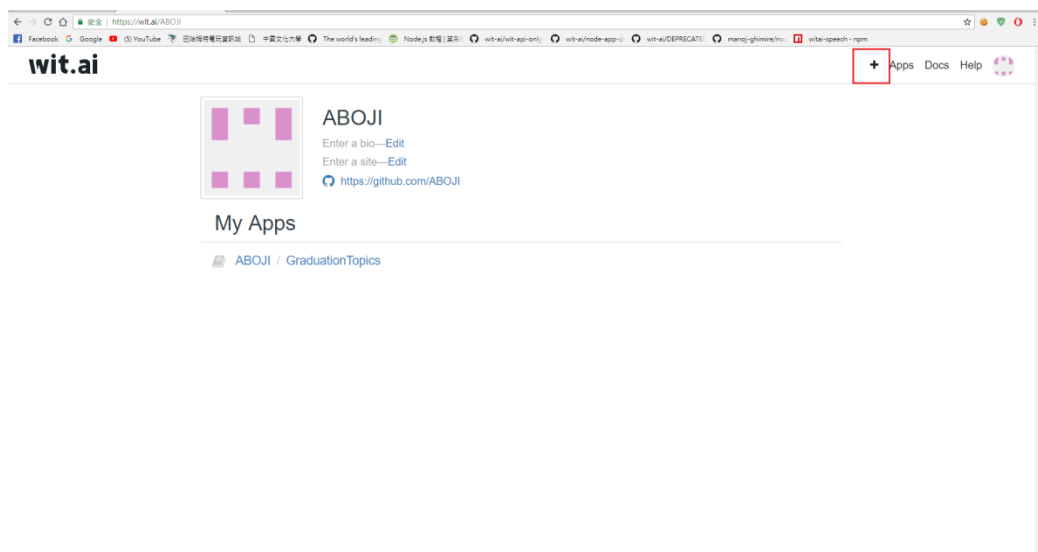


圖 3.11 AI 訓練圖

步驟 2.此為登入後介面，點擊紅色框框處創建專案，如圖 3.11。

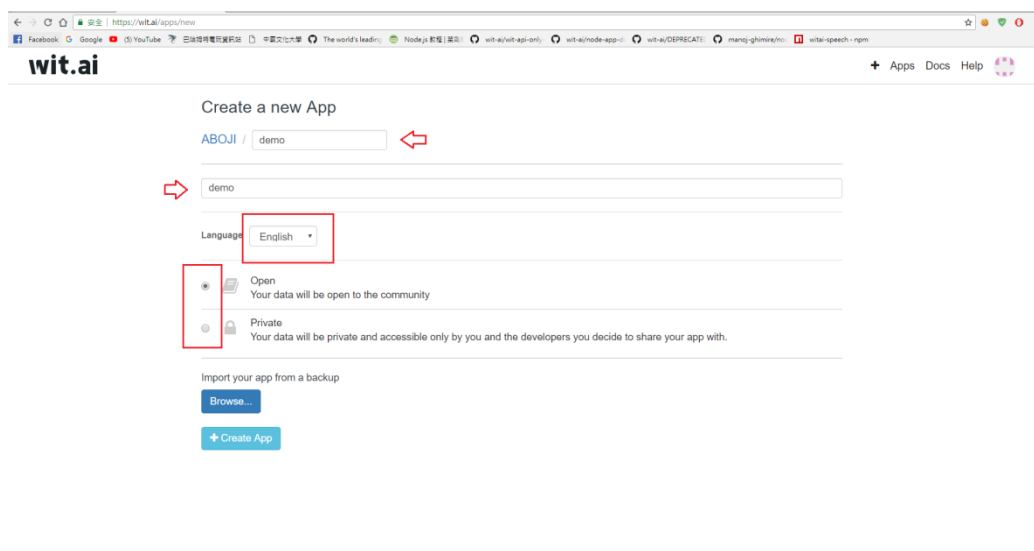


圖 3.12 AI 訓練圖

步驟 3. 此圖為創建專案的介面(箭頭分別為輸入專案名稱與描述專案，框框則為選擇語言與選擇公開或私密)，如圖 3.12。

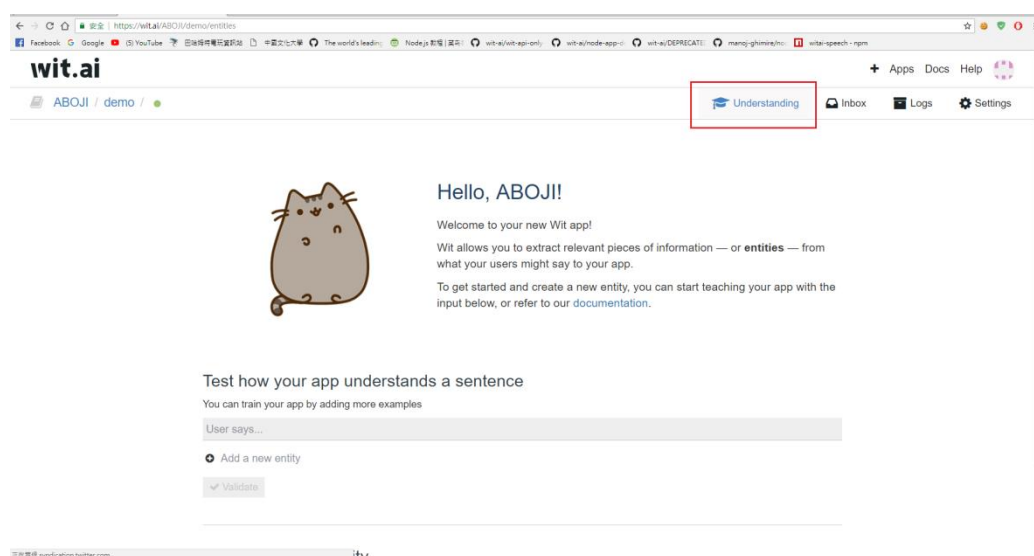


圖 3.13 AI 訓練圖

步驟 4. 此為專案介面，點擊紅色框框處”Understanding”則會跑到此介面，如圖 3.13。

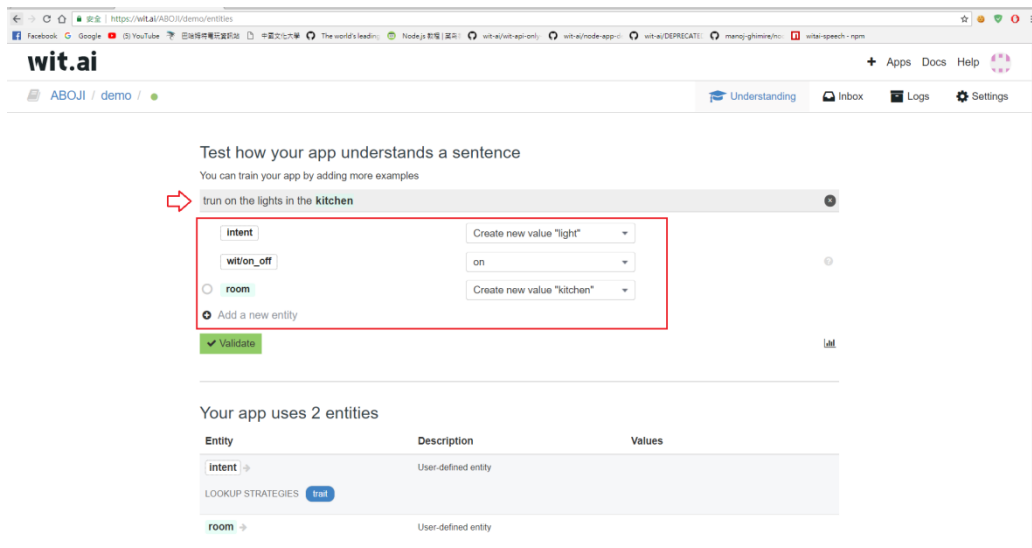


圖 3.14 AI 訓練圖

步驟 5. 於箭頭處輸入例句，並在紅色寬寬內創建或選擇相關屬性與參數，如圖 3.14。

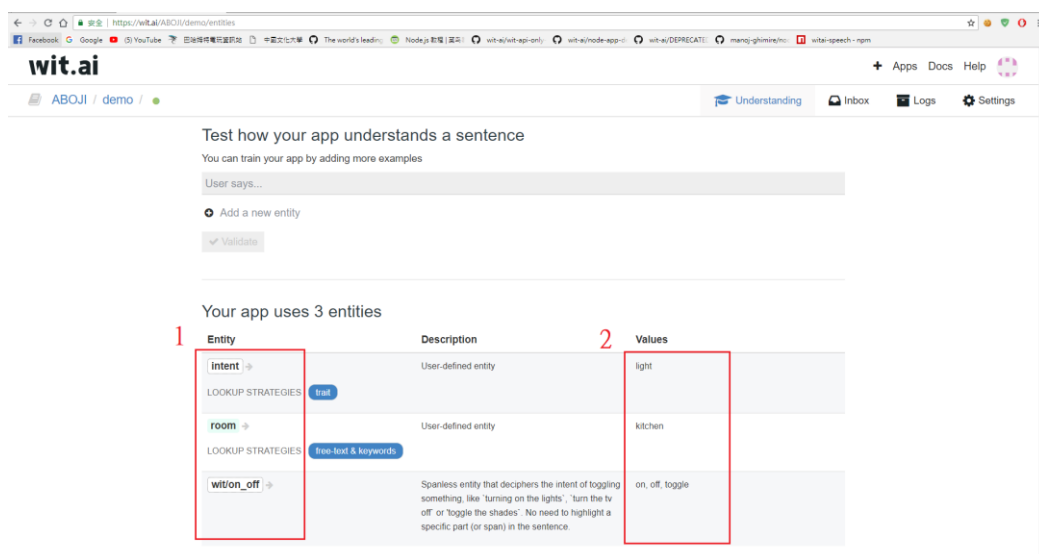


圖 3.15 AI 訓練圖

步驟 6. 此圖為輸入後的畫面，編號 1 的紅色框框呈現出的屬性與編號 2 的框框呈現參數，如圖 3.15。

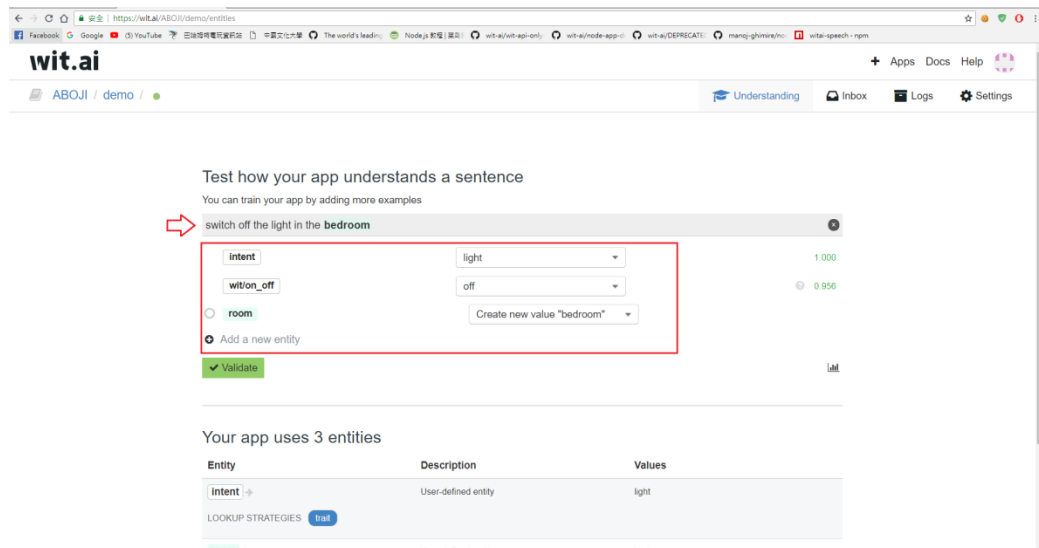


圖 3.16 AI 訓練圖

步驟 7. 再次於箭頭處輸入例句，並在紅色框框內創建或選擇相關屬性與參數，如圖 3.16。

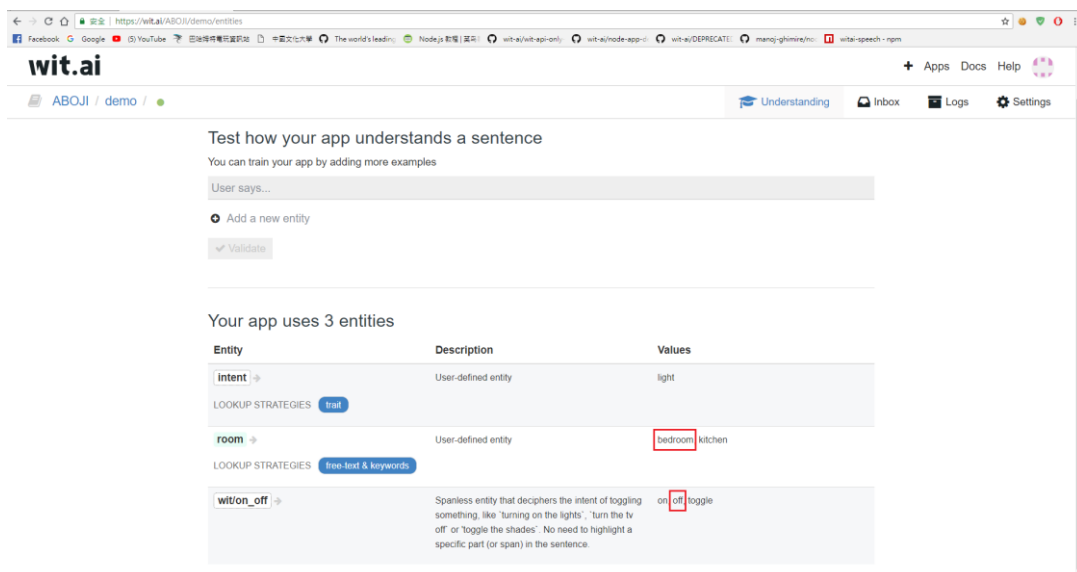


圖 3.17 AI 訓練圖

步驟 8. 此圖為再次輸入後畫面，可於紅色框框內發現新增了相關參數，如圖 3.17。

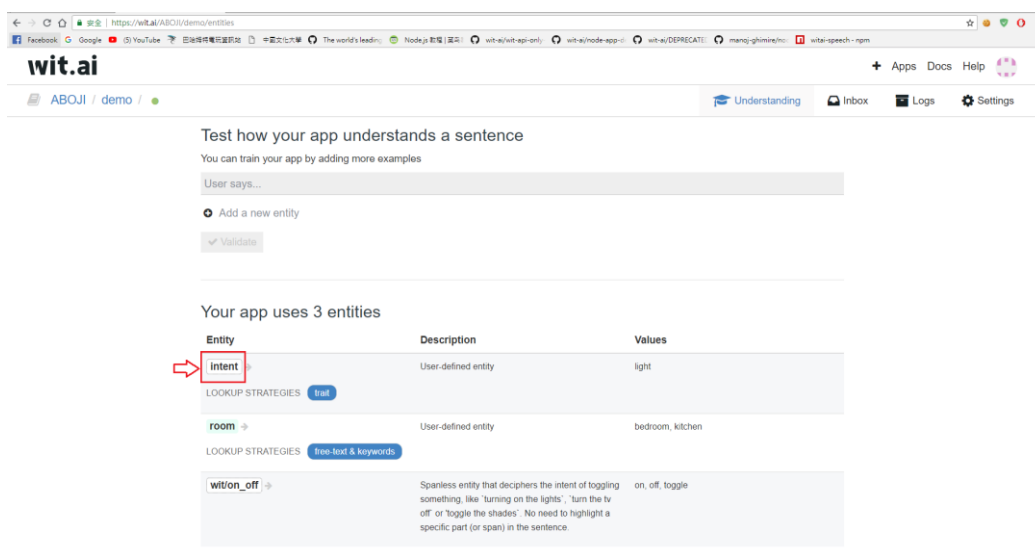


圖 3.18 AI 訓練圖

步驟 9. 直接點擊於紅色框框處 Entity 下方的相關屬性，可直接新增參數，如圖 3.18。

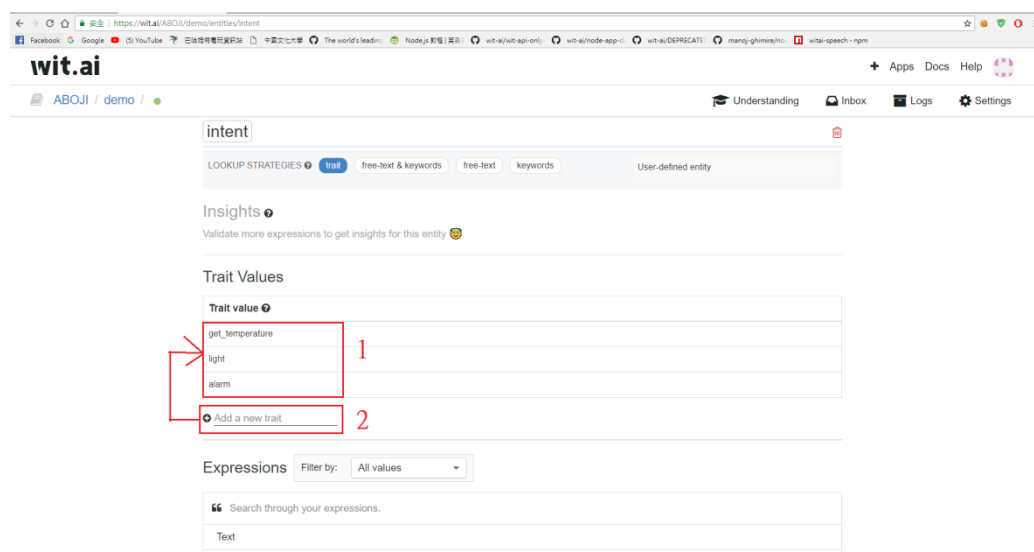


圖 3.19 AI 訓練圖

步驟 10. 此圖為 Entity 相關屬性下的介面，編號 1 的紅色框框為”當前所有的相關屬性”，編號 2 的紅色框框為”輸入要新增的參數”，如圖 3.19。

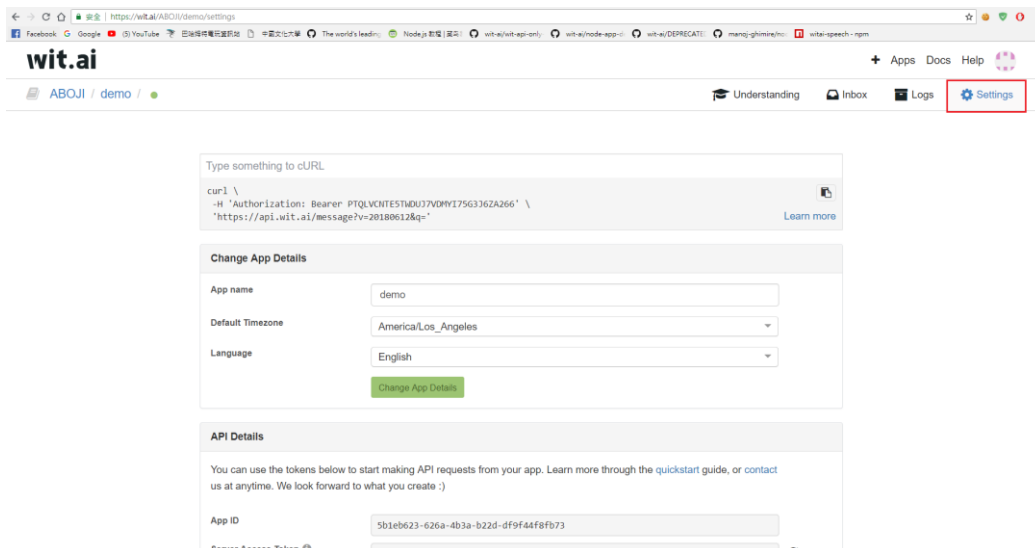


圖 3.20 AI 訓練圖

步驟 11. 此圖為設定畫面點擊紅色框框處的 Setting 則會跑到此介面，如圖 3.20。

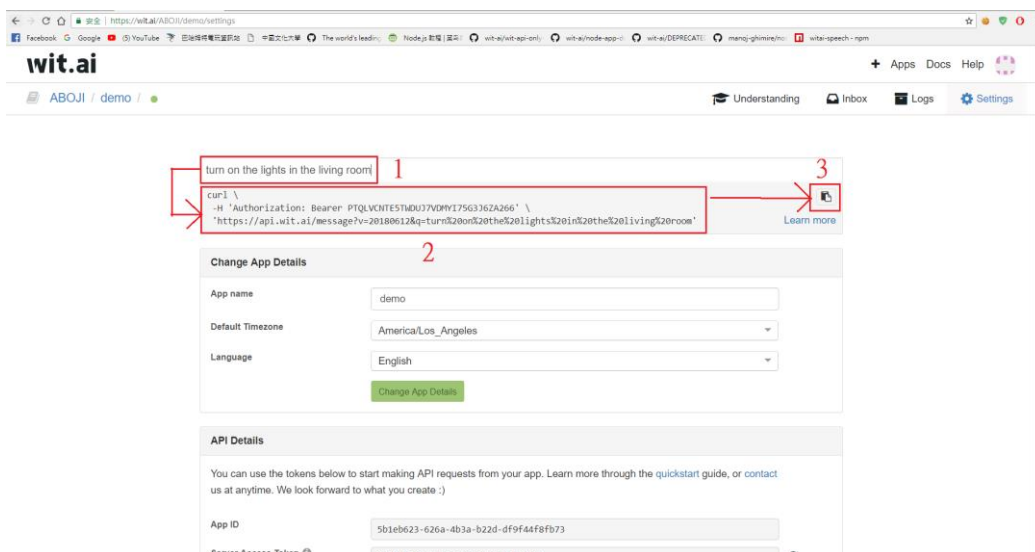


圖 3.21 AI 訓練圖

步驟 12. 於 1 號紅色框框處輸入要傳送的語句，會在 2 號紅色框框處產生程式碼，再到 3 號紅色框框處點擊複製，如圖 3.21。

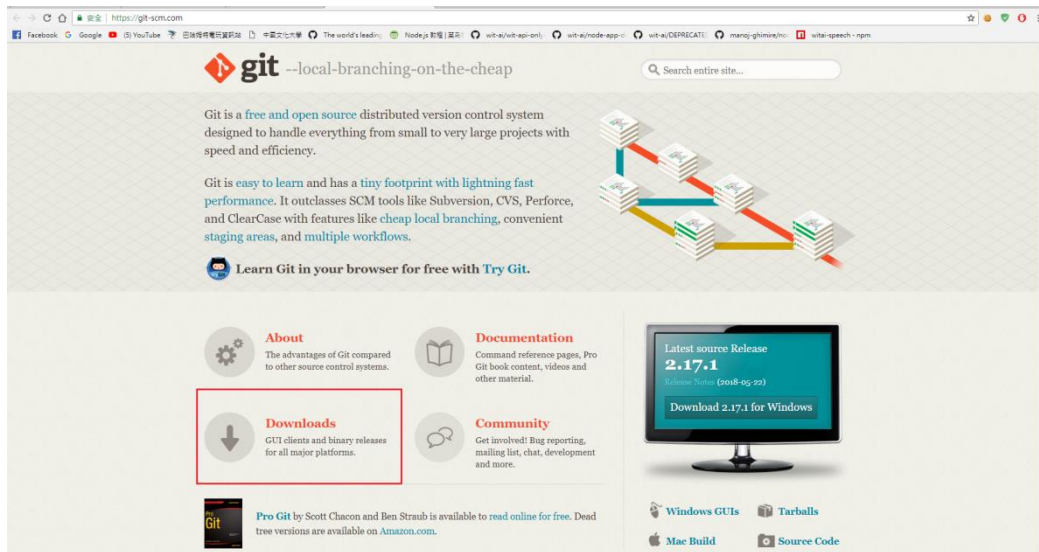


圖 3.22 AI 訓練圖

步驟 13. 安裝運行程式，前往 Git Bash(網址:http://git-scm.com)於紅色框框處點擊下載，如圖 3.22。

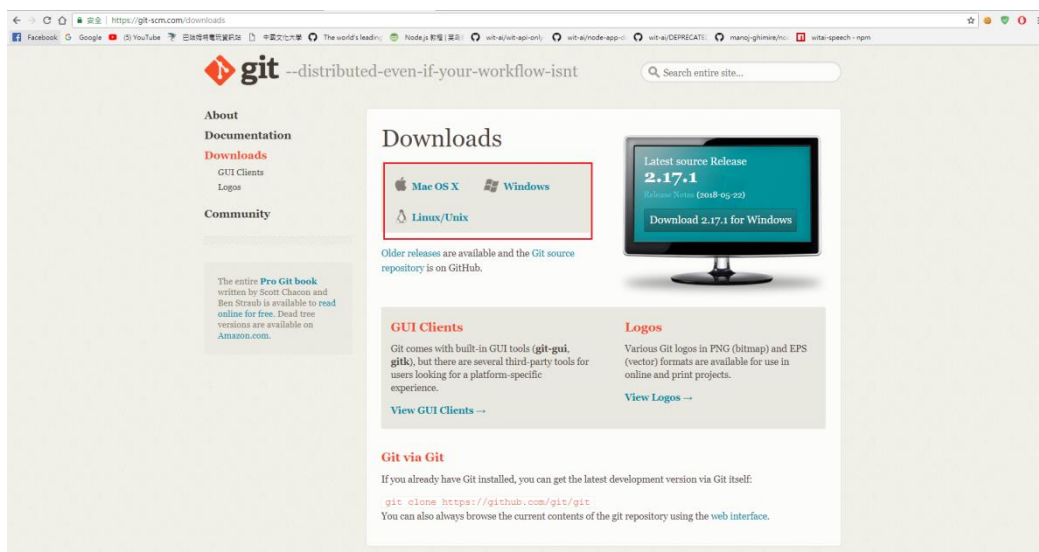
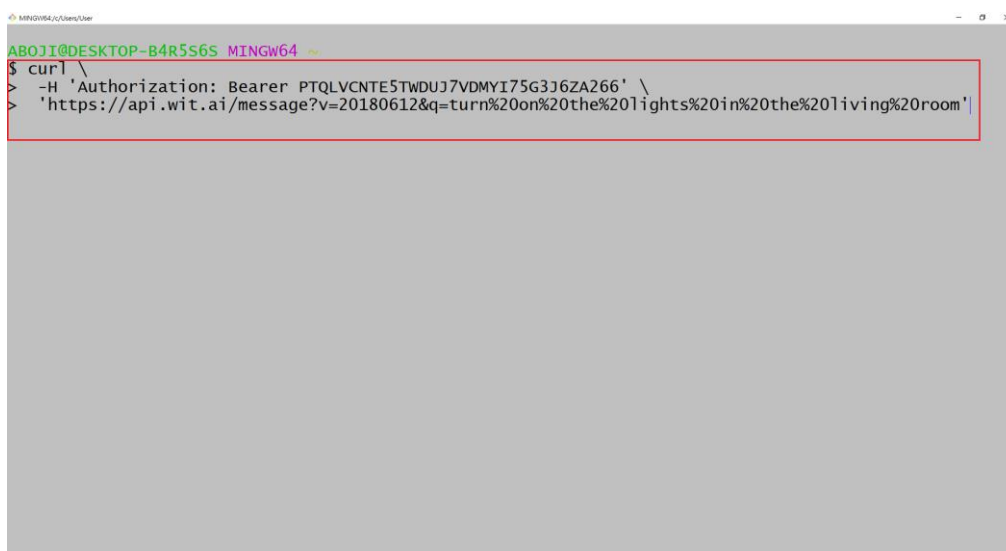


圖 3.23 AI 訓練圖

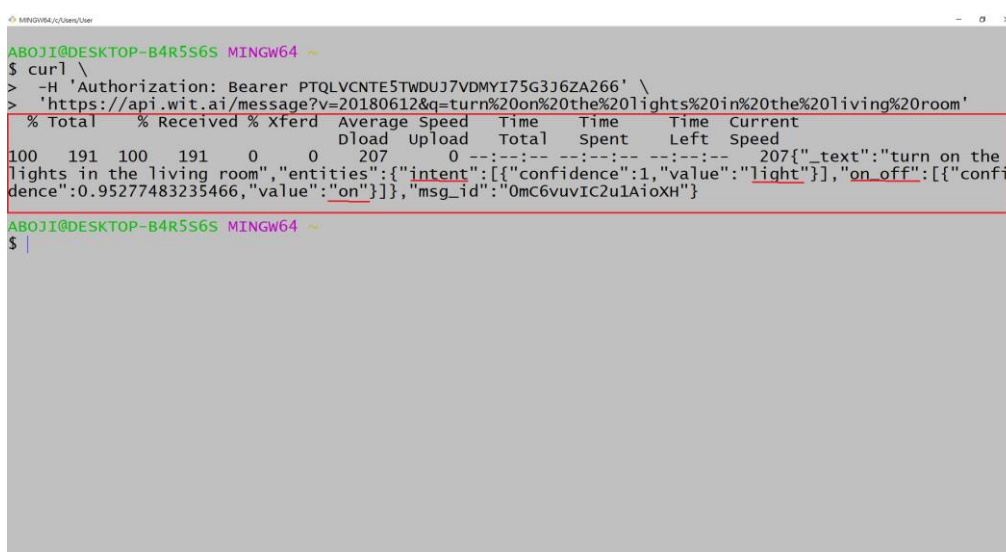
步驟 14. 依作業系統的不同，於紅色框框處選擇符合者下載，如圖 3.23。



```
ABOJI@DESKTOP-B4R5S6S MINGW64 ~
$ curl \
> -H 'Authorization: Bearer PTQLVCNTE5TWDUJ7VDMYI75G3J6ZA266' \
> 'https://api.wit.ai/message?v=20180612&q=turn%20on%20the%20lights%20in%20the%20living%20room'|
```

圖 3.24 AI 訓練圖

步驟 15. 安裝完成後開啟 Git Bash，於紅色框框處貼上程式碼，如圖 3.24。



```
ABOJI@DESKTOP-B4R5S6S MINGW64 ~
$ curl \
> -H 'Authorization: Bearer PTQLVCNTE5TWDUJ7VDMYI75G3J6ZA266' \
> 'https://api.wit.ai/message?v=20180612&q=turn%20on%20the%20lights%20in%20the%20living%20room'
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 191 100 191 0 0 207 0 --:--:-- --:--:-- --:--:-- 207{"_text": "turn on the
lights in the living room", "entities": {"intent": [{"confidence": 1, "value": "light"}], "on_off": [{"confi
dence": 0.95277483235466, "value": "on"}]}, "msg_id": "0mc6vuvIC2u1AioXH"}
ABOJI@DESKTOP-B4R5S6S MINGW64 ~
$ |
```

圖 3.25 AI 訓練圖

步驟 16. 按下 Enter 鍵後出現結果畫面，紅色框框處為回傳資料，底線標註的為屬性與參數，如圖 3.25。

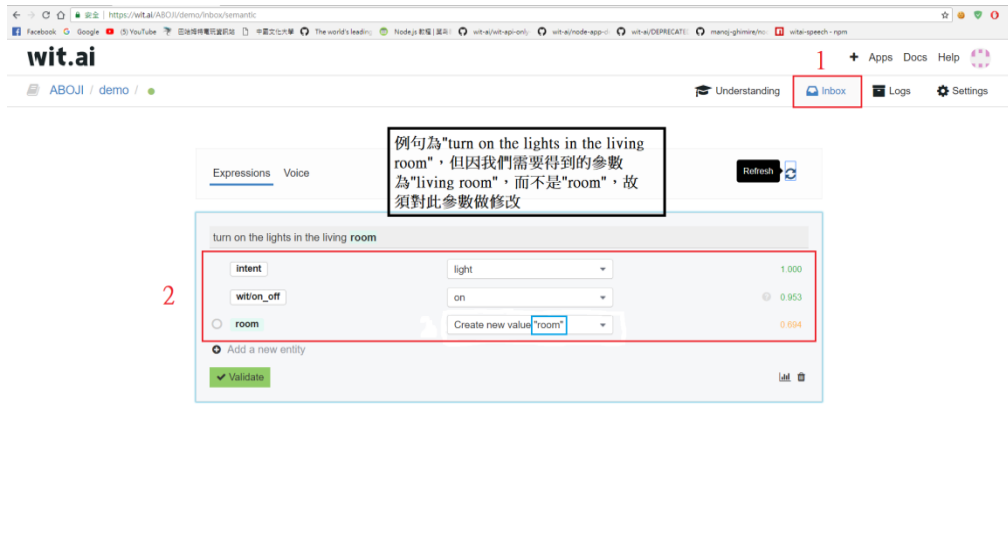


圖 3.26 AI 訓練圖

步驟 17. 回到 Wit.Ai 網站，點擊 1 號紅色框框的 Inbox，並於 2 號紅色框框內確認回傳的相關屬性與參數是否正確，若是錯誤的資料可以下拉選單進行修改，範例圖中，應顯示為”living room”而不是”room”，如圖 3.26。

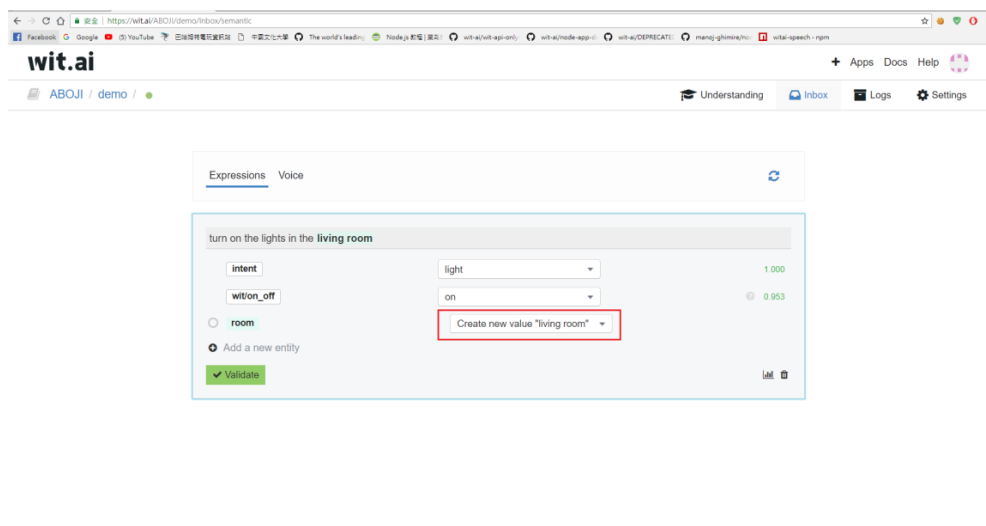


圖 3.27 AI 訓練圖

步驟 18. 此圖為修改參數後的畫面，紅色框框內為修正後的正確參數，確認無誤後點擊”validate”，如圖 3.27。

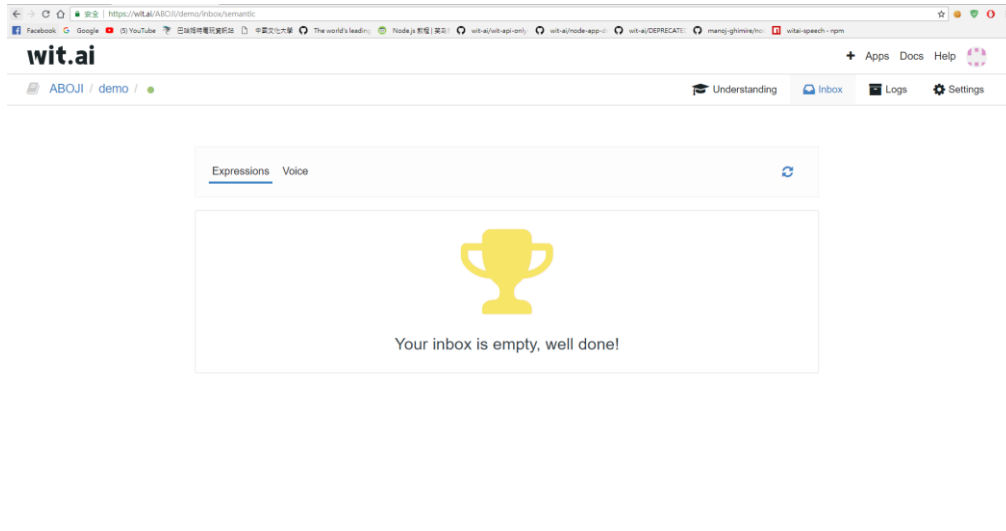


圖 3.28 AI 訓練圖

步驟 19.訓練完成，至此完成一句語音辨識訓練，如圖 3.28。

3.2.3 樹莓派系統

(1) RASPBIAN

Raspberry Pi 可以使用許多不同的作業系統，目前這些作業系統仍持續改版或增加中。其中 Raspbian 作業系統是官方推薦，也是目前最多人使用的作業系統。

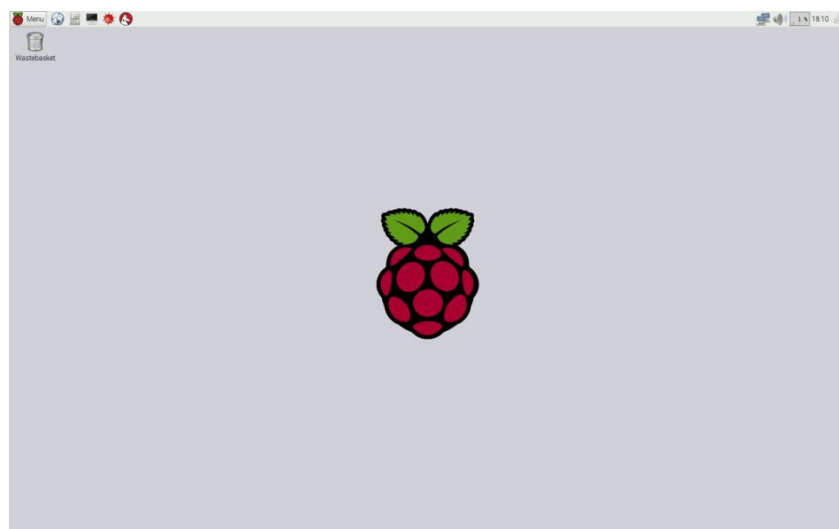


圖 3.29 Raspbian

3.3 硬體系統架構

本節將說明實驗設備、物聯網系統架構規劃，整個系統主要包含感知層、網路層與應用層。

3.3.1 系統硬體設備

本研究所使用之硬體及實驗設備如下：

- (1) 一組樹莓派(Raspberry pi 3 model B+)
- (2) 一台路由器
- (3) 一組影像感測器
- (4) 一組煙霧感測器
- (5) 一組智慧插座
- (6) 一隻 Android 手機

除了上述設備以外，還有整個實驗模型的配線、待測試的裝置部分尚未決定。

3.3.2 物聯網系統架構

本節將說明本研究使用之物聯網系統部分架構、部分採用之通訊協定與整體概念。

感知層包含了，各式各樣的感測器、樹莓派主機與行動終端。

網路層包含了，以路由器與樹莓派主機建立之外部網絡伺服器與手機終端使用的行動通訊網路。

應用層包含了，本研究之整套 APP 結合人工智慧與物聯網開發之 E 能宅-智慧家庭系統。

。

(1) 物聯網系統架構

透過區域劃分，方便管理所有裝設之設備，採用的設備多數採用 ZigBee 協定，透過內部網路連線至樹莓派主機，在使用路由器將樹莓派指定為固定 IP 後架設外網伺服器，至此行動用戶端將可透過外網無視空間限制，即時掌握家中環境，如圖 3.30。

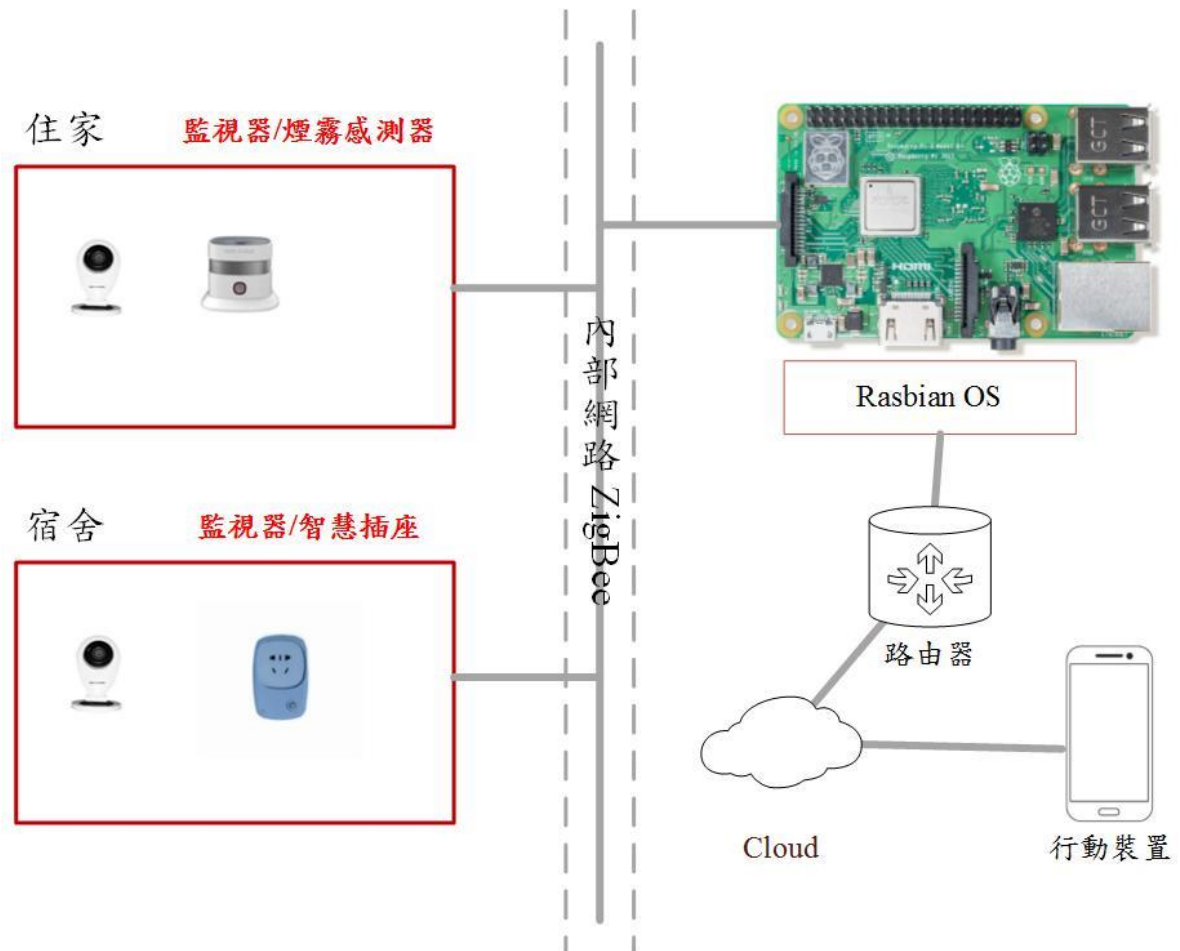


圖 3.30 物聯網系統概念示意圖

第 4 章 初步實驗結果與人力配置

4.1 研究進度

本節將說明當前研究進度，主要分為 APP 系統、人工智慧系統與物聯網系統三大部分，以下將說明各項工作進度與成果呈現。

4.1.1 手機 APP 部分

目前已確定使用 Android Studio 與 JAVA 撰寫當開發環境，初步的 GUI 構思與架構已經擬定完成，下一個目標會全心投入程式碼解決製作上的問題，這部分還需要量時間解決，也會去涉獵關於美編的知識將我們的 GUI 作的更完美。

(1) APP GUI 設計

以下為我們大致上的 GUI 框架，目前是以 VISIO 先做出大致上的框架模式，確認功能也按鈕位置等等，日後會再以 PhotoShop 重製，並套入 Flow 所提供的模組，並且呈現出一些動態效果。



圖 4.1 APP 登入頁面

此圖為我們的登入頁面，日後設計出 LOGO 圖以後會再加入。以簡潔的介面為目標，設計書讓使用者舒適的使用環境，如圖 4.1。



圖 4.2 APP 主控制頁面

此圖為我們的主控制頁面，這邊主要的功能為讓使用者能夠新增區域，我們想要讓使用者將各個區域的設備集中再一起，例如將宿舍的設備放在一塊，家中的設備放在一塊，以方便在未來找尋與使用，如圖 4.2。



圖 4.3 APP 新增區域頁面

此圖為我們的新增區域頁面，再使用者點及新增以後，會來到這個畫面，並讓使用者自定區域名稱，如圖 4.3。



圖 4.4 APP 區域配置頁面

此圖為我們的區域配置頁面，此頁面功能較多，主要功能為語音控制，使用者在此頁面典籍下方語音控制按鈕，便可輕鬆控制該區域中的所有設備，左側放大鏡則是用於查看設備當前狀況，右側垃圾桶為刪除設備按鈕如，圖 4.4。



圖 4.5 APP 裝置設定頁面

此圖為我們的裝置設定頁案，該頁面用於讓使用者能夠得知設備當前狀況，對於不同設備會有不同的資料介面，此處以溫溼度感測器為例子，會顯示對應設備的詳細資訊如，圖 4.5。



圖 4.6 APP 選單頁面

此圖為我們的選單頁面，在首頁頁面時由左向右滑出便會出現該選單，主要呈現出該 APP 所有設定內容如，圖 4.6。



圖 4.7 語音辨識頁面

此圖為我們的語音辨識介面，再開啟語音辨識功能時，就會跳到這個頁面，用於聽取使用者指令，圖中波浪會透過 Flow 給予他波浪效果，如圖 4.7。

4.1.2 人工智慧部分

當前研究進度為了使 Wit.ai 提供的套件得以運用，在此先建立了可套用的程式平台 Node.js，並試著在此系統中建立起測試用的環境，在下個階段會試著在此程式中運行語音辨識的流程，並與手機 APP 系統建立起連結，關於這部分的資料還需繼續研究，才可使 AI 系統更為完整，以下將說明 node 環境設置流程。

(1) node 環境設置流程

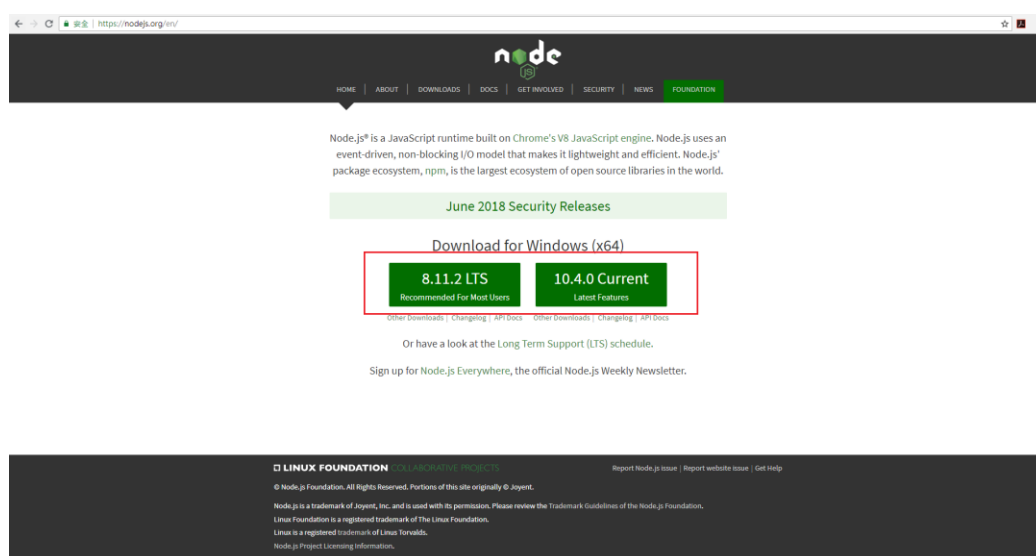


圖 4.8 node 環境設置圖

步驟 1. 前往 Node.js 網站(<https://nodejs.org/en/>)並在紅色框框處中選擇你要安裝的版本號，如圖 4.8。

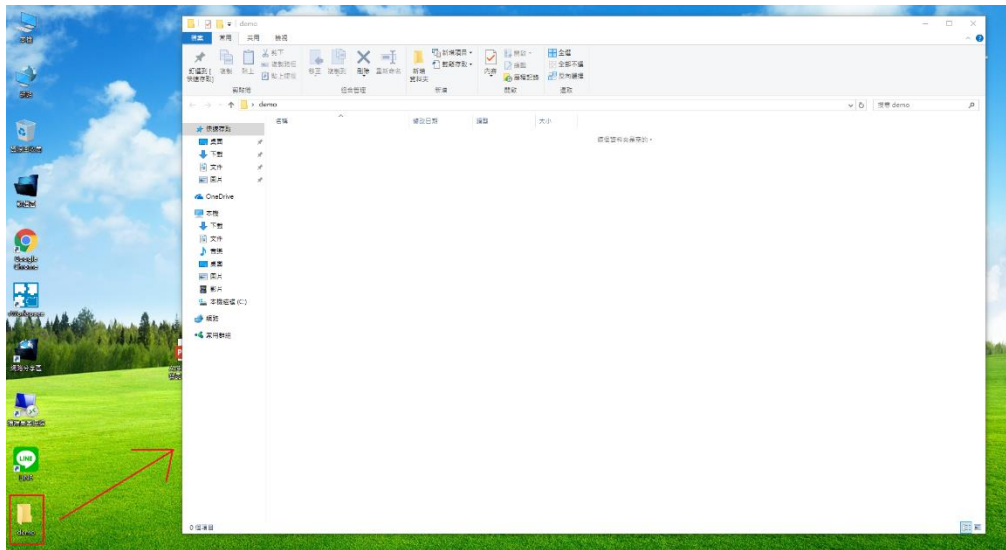


圖 4.9 node 環境設置圖

步驟 2. 在想要的路徑中創建專案資料夾，這裡則是在桌面上建立紅色框框內的新資料夾，名稱叫 demo，如圖 4.9。

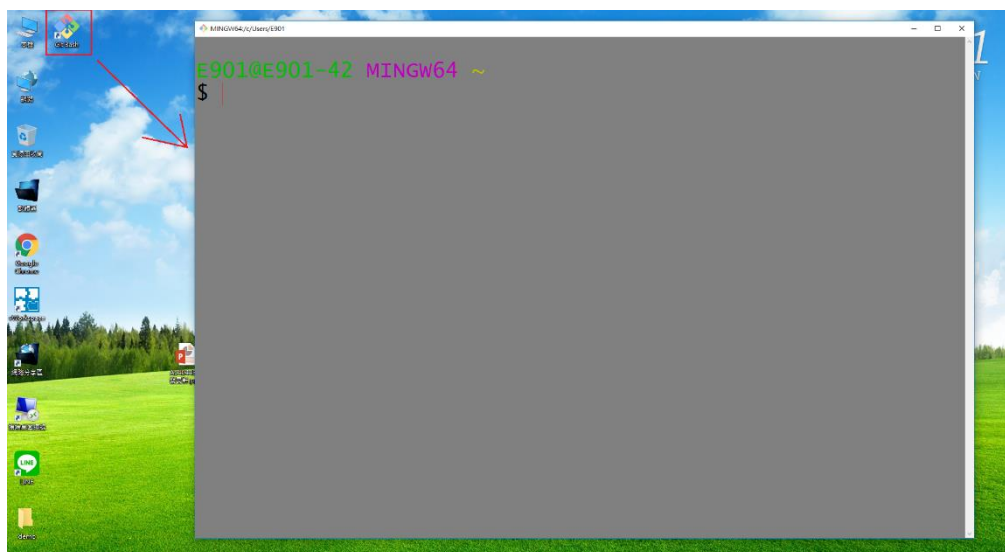


圖 4.10 node 環境設置圖

步驟 3. 開啟 Git Bash，如圖 4.10。

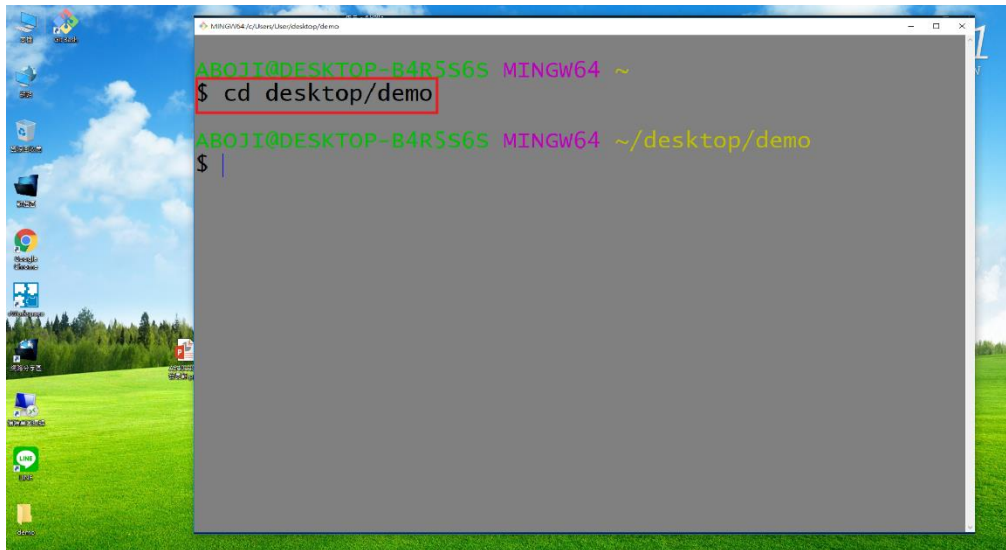


圖 4.11 node 環境設置圖

步驟 4. 在 Git Bash 中輸入紅色框框內的程式碼後，點擊 Enter 鍵，即可移至紅色框框中所指示的路徑，而這個路徑即為你的專案資料夾所在的路徑，如圖 4.11。

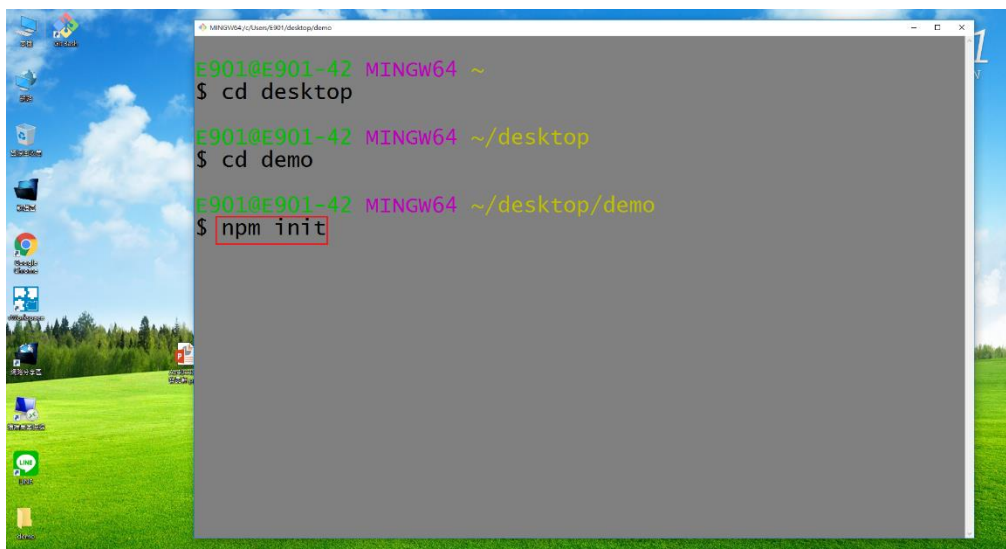


圖 4.12 node 環境設置圖

步驟 5. 在 Git Bash 上輸入紅色框框中的程式碼後，點擊 Enter 鍵，即可開始環境的初始化，如圖 4.12。

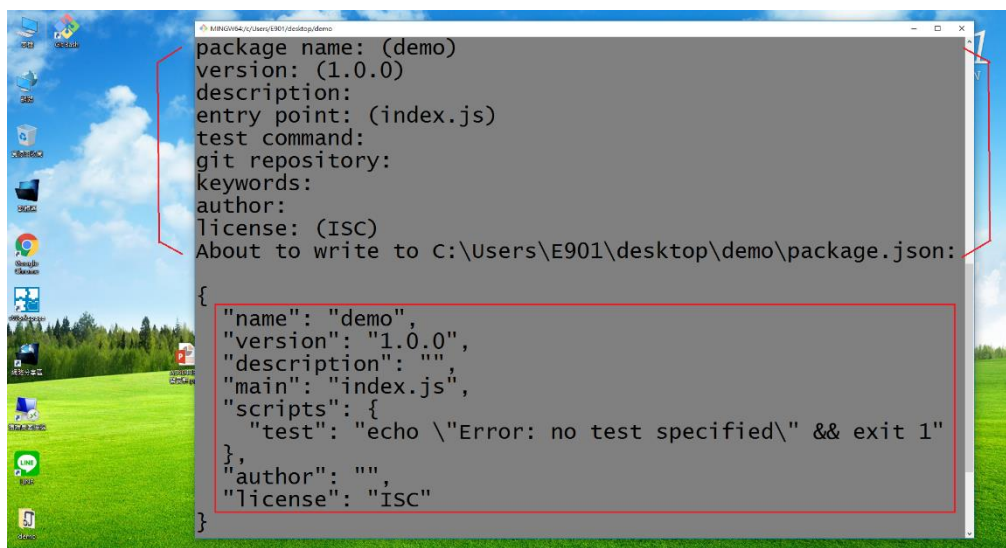


圖 4.13 node 環境設置圖

步驟 6. 在紅色括弧中輸入所要求的資料後，點擊 Enter 鍵，將在紅色框框的位置顯示出剛剛輸入的資料，確認無誤後，點擊 Enter 鍵，即可完成環境的初始化，如圖 4.13。

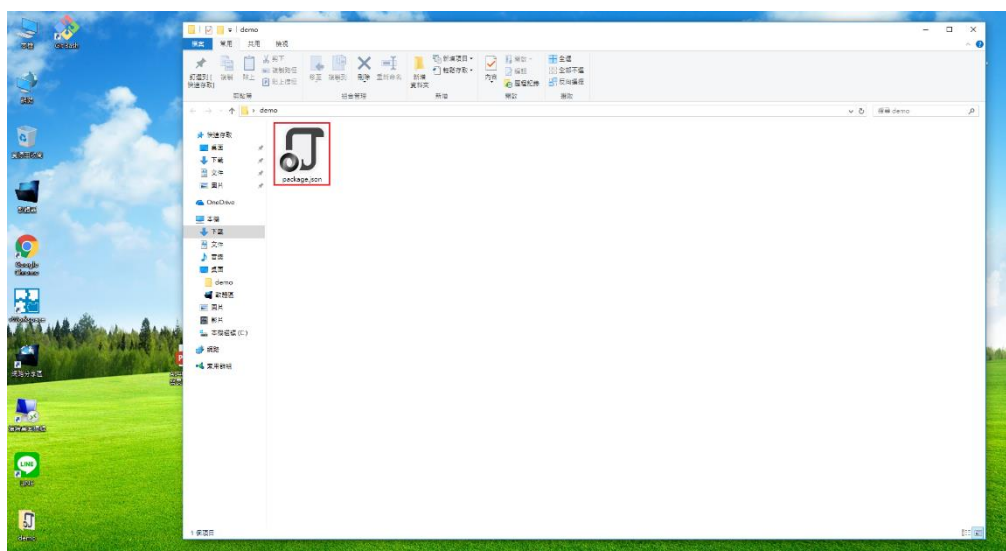
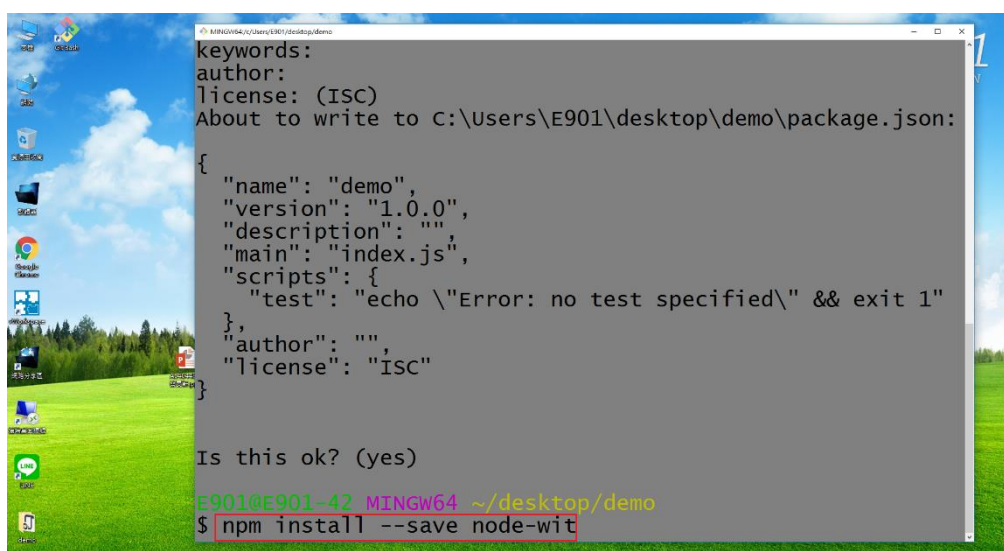


圖 4.14 node 環境設置圖

步驟 7. 完成後，確認專案資料夾內是否建立起紅色框框中的 package.json 檔，這個檔案將會顯示出已經安裝的套件，如圖 4.14。

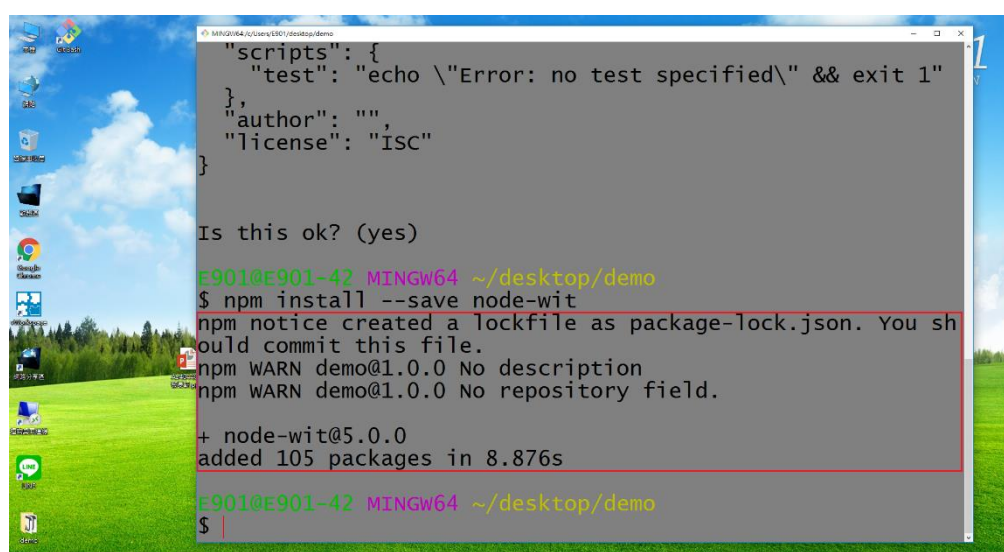


```
keywords:
author:
license: (ISC)
About to write to C:\Users\E901\desktop\demo\package.json:
{
  "name": "demo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this ok? (yes)
E901@E901-42 MINGW64 ~/desktop/demo
$ npm install --save node-wit
```

圖 4.15 node 環境設置圖

步驟 8. 開啟 Git Bash，輸入紅色框框中的程式碼後，點擊 Enter 鍵，即可將需要的套件加入至 package.json，如圖 4.15。



```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this ok? (yes)
E901@E901-42 MINGW64 ~/desktop/demo
$ npm install --save node-wit
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN demo@1.0.0 No description
npm WARN demo@1.0.0 No repository field.

+ node-wit@5.0.0
added 105 packages in 8.876s
E901@E901-42 MINGW64 ~/desktop/demo
$
```

圖 4.16 node 環境設置圖

步驟 9. 若成功加入，則會在 Git Bash 中顯示出紅色框框內的程式碼，如圖 4.16。

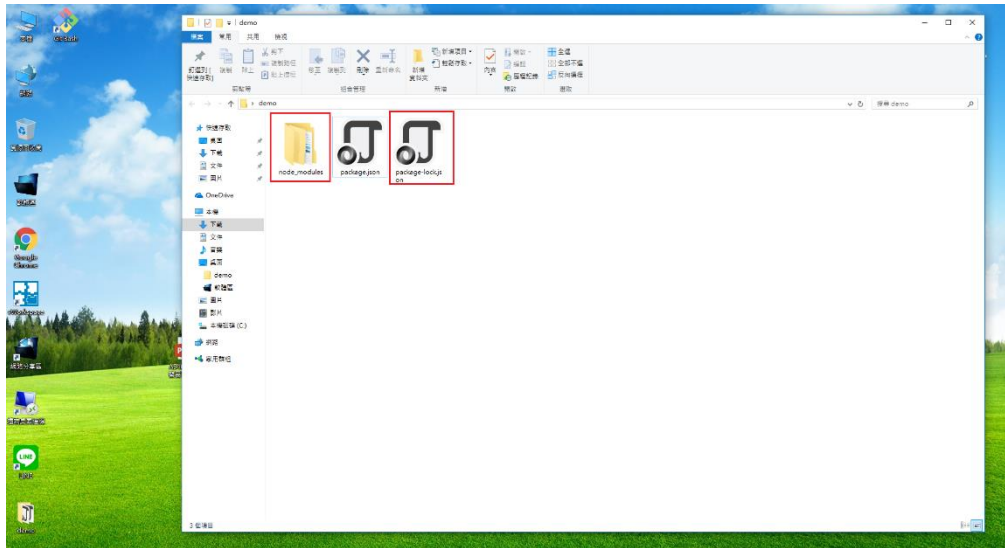


圖 4.17 node 環境設置圖

步驟 10. 成功加入後，專案資料夾內則會建立起紅色框框中的資料夾與 json 檔，如圖 4.17。

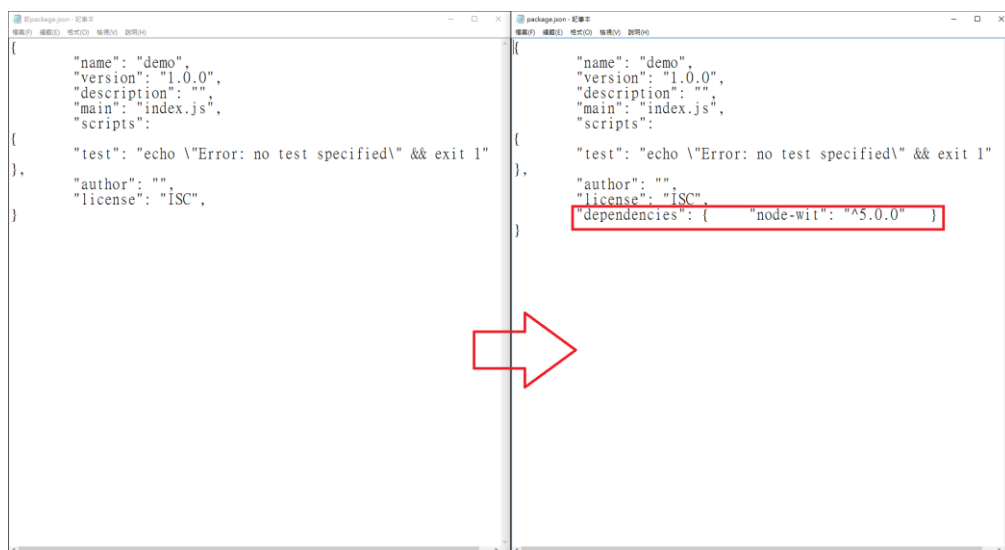


圖 4.18 node 環境設置圖

步驟 11. 加入套件後，專案資料夾內的 package.json 檔中的內容將會從左邊的程式碼變為右邊，而兩者之間的差異在於多了紅色框框中的程式碼，如圖 4.18。

4.1.3 物聯網部分

當前研究已確定使用 Raspberry pi 3 model B+作為核心主機，並採用 Raspbian 作為 OS 系統，現已完成樹莓派系統建置、各種感測器的選用及採買與物聯網部分實驗模型初步設計。

因現階段查到的物聯網資料多數都是透過內網的方式建立連線與管理，但此方法與本研究之核心方向不相符，因此未來預計先去確立外網伺服器具體架設之技術，成功架立後再回過頭連接感測器等設備，以下將展示現已完成之樹莓派系統建置。

(1) 樹莓派系統建置

透過 Win 32 Disk Manager 燒錄最新版本 Raspbian 到 Micro SD 卡後，開啟 Terminal 進行系統初始設定，在輸入指令令軟體更新至最新版本，而後加入中文輸入法等調整，如圖 4.19。

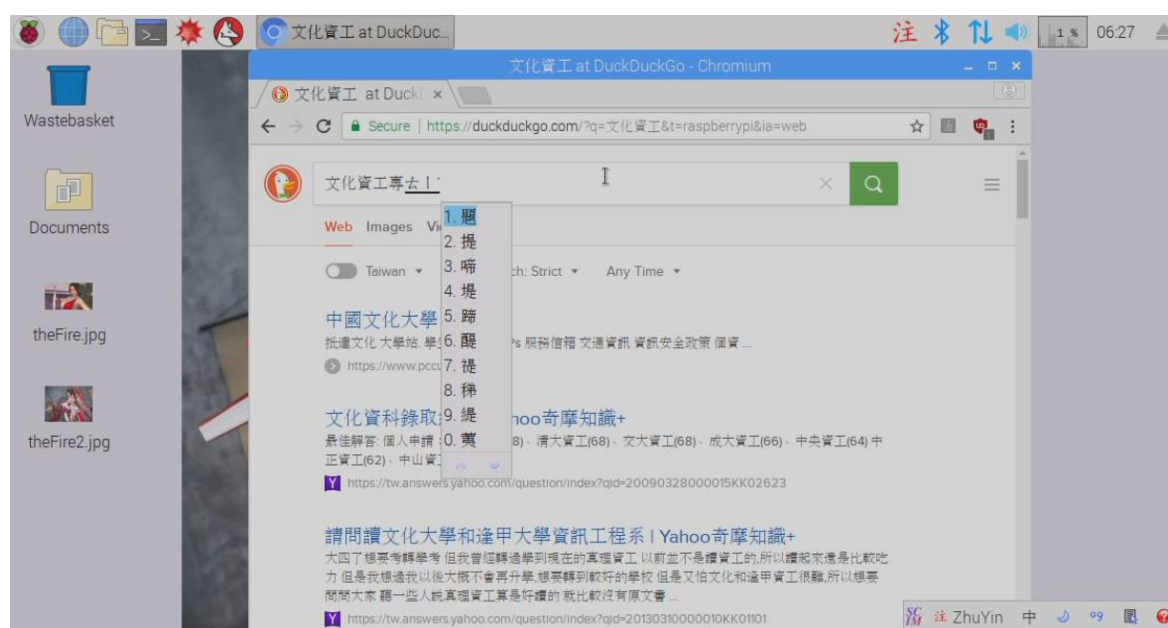


圖 4.19 Raspberry pi 運行截圖

4.2 研究人員分配

由一般居家環境正常使用之概念架構，選擇以居住家庭及辦公室為個案，發展出可行的居家管理 APP。

參與本研究之工作人員及預定之工作內容如表 4.1 所示。

表 4.1 參與本項研究之人員及工作內容

姓名	工作內容
吳梓境 紀俊丞	介面規劃設計、程式設計、程式開發督導
吳承諺	樹梅派硬體架構之設計、各種感測器規劃 專題統整規劃、物聯網系統設計
蔡旻軒	人工智慧應用、WIT.AI 程式設計與實行

第 5 章 預期成果

由一般居家環境正常使用之概念架構，選擇以居住家庭及辦公室為個案，發展出可行的居家管理 APP。

本研究規劃案，預期可以得到下列之成果：

(1) 便利性

以往家庭中，所以電器都有獨立開關，不論按鍵式或是遙控式，希望透過我們所設計的 APP 來給予使用者們更多的方便性，將所有電器的控制方式移植至手機上，甚至能夠透過語音操控，以獲取更大的便利性。

(2) 等待時間的減少

以往任何電器設備都幾乎都有一些等待時間，如冷氣開了需要一段時間房間溫度才會明顯下降，又或者是將熱水煮開的這段時間等等，希望能夠透過 APP 的遠端以及定時控制，有效的減少使用者在使用這些電器產品時的等待時間，可以在預定時間開啟某項產品，例如：能將開水設定在早晨幾點鐘自動煮水，起床後能夠直接使用，不須再等待燒開水的這段時間。

(3) 弱勢族群

我們 APP 希望能夠做到完全透過手機控制所有產品，對於行動不方便的族群，不用再請他人幫忙拿取遙控器之類的，又或者是簡單的按鍵式開關都能夠透過手機實行，我們也將寫入語音方面的功能，對於任何族群都將達到方面性的提升。

參考文獻

- [1] 鄭永楨, “International Journal of Science and Engineering” ,台南大學,vol.5, no.1,pp.17-22
- [2] 王路敬, “電腦編成技巧與維護” ,中國信息產業商會,vol.18, pp.44-46, 2017
- [3] 李國森, “信息通信” ,湖北通信服務公司,vol.8, pp.115-116, 2015
- [4] 張欣, “科技資訊” ,北京國際科技服務中心,vol.15, pp.38-40, 2016
- [5] “智能感測器”,2018, May 18, In Wikipedia the free encyclopedia, Retrieved June 12,2018,<https://zh.wikipedia.org/wiki/%E6%99%BA%E8%83%BD%E6%84%9F%E6%B8%AC%E5%99%A8>
- [6] “物聯網”,2018, June 11, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://zh.wikipedia.org/wiki/%E7%89%A9%E8%81%94%E7%BD%91>
- [7] “語音識別”,2018, May 17, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://zh.wikipedia.org/wiki/%E8%AF%AD%E9%9F%B3%E8%AF%86%E5%88%AB>
- [8] “Android studio”,2014, Dec 12, In Wikipedia the free encyclopedia, Retrieved June 13,2018,<https://www.ithome.com.tw/news/92892>
- [9] “樹莓派”,2018, June 6, In Wikipedia the free encyclopedia, Retrieved June 13,2018<https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%B5>